

SyncWare News

The journal for technical
applications of T/S
computers

Volume 4 Number 3

Jan.-Feb. '87



T/S On Line

Probably the least tapped resource by the average TS user is telecommunications (see SWN 4:1). In my case, the problem is primarily one of not having enough information about the subject to comfortably dive in and start communicating. Thanks to the efforts of Pete Fischer, Box 2002, Temple, AZ 85281, and Steve Ishii, 18414 Deloise Ave., Cerritos, CA 90701, anyone who had been hesitant to enter the world of telecommunications no longer has an excuse to take the plunge.

Messrs. Fischer and Ishii have assembled and distributed to every TS User Group they know of, a public domain work entitled "THE GUIDE TO T/S TELECOMMUNICATIONS". The User's Groups have been instructed to photocopy and distribute this guide to their membership. If you haven't seen a copy yet, get on your group's President or Secretary. If they can't help you, send \$2.00 to either Mr. Fischer or Mr. Ishii to cover copying and postage, and they will send you your own personal copy.

This definitive guide to T/S related Bulletin Board Service(s) (BBS), explains everything from BBS etiquette and how a BBS works for the novice user to downloads, advanced usage, and how to start your own BBS for the experienced user. Also covered are T/S hardware and software, hardware and software modifications, and, of course, the names, phone numbers, and protocol for the many BBS currently catering to the T/S line of computers.

Planned for release in early 1987 is the deluxe edition of the book which will be in an 8-1/2" x 11" loose leaf format. It will include all the text of Version 1.0, plus more Boards, complete help files from the popular Boards, listings of available downloads, info on international telecommunications, any corrections received on Version 1.0, and more. The cost for the new deluxe version will be \$5.00.

Now is the time to get on the telecommunications band-wagon! TS2050 modems are still relatively easy to come by. Several manufacturers are offering modems and software of equal or better quality than the 'officially' supported Timex equipment. A source like Fischer and Ishii's "THE GUIDE TO T/S TELECOMMUNICATIONS" is all you need to tie it all together.

FOR YOUR SUPPORT

This column announces any software, hardware, and related modifications that are new or otherwise untested by us. If you have something of interest, we will announce it here. Send us a description of your product (PLEASE keep it short). We advise readers to send a self addressed and stamped envelope, as a courtesy, to the individuals or companies for information.

Fred Nachbaur reports that DUNGEON OF YMIR V3, is complete. This version runs on a ZX81/TS1000 with 8K NVM, and uses the hi-res system published in SWN 4:2. This is a challenging and complex D&D-style adventure game with hi-res graphics, written entirely in machine-code. Cost is \$24.95 postpaid.

The John Oliger Co., 11601 Whidbey Dr., Cumberland, IN 46229, is currently offering the Oliger 2068 Disc System. Available as bare boards, kits, or completed and tested units, prices range from \$17.95 to \$119.95 pp. The disk system consists of two boards which plug into the Oliger 2068 Expansion Board (not included). Features include ease of use, high speed, and Spectrum and TS2068 compatibility. Write for complete details and price info.

Grey & Clifford Computer Products, P. O. Box 2186, Inglewood, CA 90305, (213) 759-7406, is offering software and hardware for communications on the Spectrum emulated TS2068. SPECTERM-64 is terminal emulation software which allows 64 column display without additional hardware. Price for SPECTERM-64 is \$30. When used with the new Z-SI/O hardware card, it allows use with any RS-232C modem or peripheral including Hayes compatibles. Price for the Z-SI/O is \$79.95. Write for more details.

Larry Kenny, Larken Electronics, has announced that he is working on a cart-bases DOS for the Larken 2068 disk drive. Features include Spectrum compatibility, built-in NMI "snapshot" save, usage of standard tokens, and disk compatibility with other systems. Also in the works are EPROMs for use with other popular disk systems. Contact LARKEN ELECTRONICS, RR#2, Navan, Ontario K4B 1H9, Canada, to be placed on the "inform-when-ready" list, or to suggest features that YOU would like to see.

Chia-Chi Chao, 73 Sullivan Drive, Moraga, CA 94556, announces that he is now selling three programs from JRC Software. Diamond Mike II, \$17.00; Great Game & Graphics Show, \$17.00; Compass, \$19.00. All prices include shipping. The games are available on tape or 5.25" Aerco disk. A complete catalog is available with SASE.

Robert C. Fischer, producer of PRO/FILE EXTENSIONS, T/S GRADER, WORD PUZZLER, and WORD GAMES, has changed his address and can now be found at Rt. 2 Arizona St., Emerson, GA 30137

Bill Bell, 596 Cherrington Road, Westerville, OH 43081, (614) 882-3883, is offering BBDOS for the TS1000 equipped with the Aerco FD-ZX Disk Interface. BBDOS is a fully automatic, BASIC transparent, disk operating system that locates itself in the 8-16K region of RAM. It is advertised as being fully compatible with all printer interfaces. \$29.95.

G. Russell Electronics, RD 1 Box 539, Centre Hall, PA 16828, (814) 364-1325, has their current catalog available upon request. Many hardware and software items are listed for the QL, 2068, and 1000 computers.

~~~~~ DUNGEON OF YMIR ~~~~~

A MULTI-LEVEL MAZE ADVENTURE GAME by Fred Nachbaur (C)1986

FOR THE TIMEX TS1500

FINALLY! A FULL-FEATURE, HIGH RESOLUTION DUNGEON GAME FOR THE TS1500!

This 24K game, written entirely in machine-code, is the most spectacular program ever written for the TS1500. Nine levels, 16 types of monsters, 14 objects, six spells. Easy to play, difficult to master. Includes FAST-SAVE with auto-boot to save in-progress games; time to load entire program reduced to 70 seconds! Revolutionary TRUE HI-RES puts your TS1500 on a par with much larger machines.

Send \$24.95 (cheque or MO) to FRED NACHBAUR, C-12 MTN. STN. GROUP BOX, NELSON BC V1L 5P1 CANADA. Specify version: V1 (TS1500 + 8K Hunter NVM or equivalent) or V2 (TS1500 + 16K RAM pack). V2 requires a minor hardware addition (included). COMING SOON: V3 for ZX81/TS1000. Inquire. *** ALSO AVAILABLE: TS1500 HI*RES EXTENDED BASIC (\$16.95)

~~~~~ DUNGEON OF YMIR ~~~~~

A MULTI-LEVEL MAZE ADVENTURE GAME by Fred Nachbaur (C)1986

-- THE CAST OF CHARACTERS --

\* - THIS IS YOU, the Hero in this saga. You must find

\* - THE SWORD OF KASLO, the object of your sacred quest.

\* - THE ORACLE; perhaps he'll help you, perhaps not....

\* - Along your way, you will encounter many strange things

~~~~~ MONSTERS ~~~~~

* - MINOTAUR OF CASTLEGR

* - ARCHER OF ARGENTA

* - DRAGON OF IMPERANCE

* - BIRD WOLF OF SILVERING

* - THREE-LEGGED GRENIN

* - GIANT KILLER COCKROACH

* - OBSCURIOUS BATTLE-BAT

* - GHOSTLY GOAT GROOM

~~~~~ TREASURES ~~~~~

\* - MAGICAL SPELL VIAL

\* - PSYCHIC LAMPLIGHT

\* - CHEST OF MYSTERY

\* - STRIPS OF - POOR

\* - SACRED TEMPLE OF ASTUM

\* - GOLD TEMPLE OFFERING

\* - PIT - CEILING HOLE

\* - 100 GOLD PIECES

~~~~~ THE CONTROLS ~~~~~

S - Move through maze

D - Cast a Destroy Spell

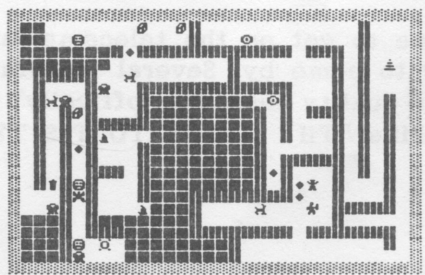
S - Cast a Shield Spell

H - Take a Healing Potion

T - Cast a Teleport Spell

R - Torment Rejuvenation

ALL
ARTWORK
IN THIS AD
FROM ACTUAL
SCREEN
DUMPS!



A Trapped Gremilin

Hit: 6 | Exp: 303 | G: 5 | L: 1 | S: 4 | R: 0 | B: 1 | Heal: 0

By arrangement with Ray Kingsley, Fred Nachbaur (C-12, Mtn. Stn. Group Box, Nelson BC VIL 5P1) has a version of HOT Z II relocated entirely into 32-48K. This allows full 16K programs to be disassembled. Requires M1 NOT mod to run m/c in high memory. Licensed Hot Z II users (only) can get this version for \$10 postpaid.

The EZRA Group II, P.O. Box 5222, San Diego, CA 92105 is currently offering inexpensive programs and custom programs in listing form. They provide programs for the ZX81, TS1000, TS1500, TS2068, and a host of other makes.

RMG Enterprises, 1419 1/2-7th Street, Oregon City, OR 97045, announces SEKTOR 2068 for the FD-68 user. This utility program is an user friendly sector editor for the 2068 and Aerco FD-68 disk system. Features include Read/Write/Correct Sectors, Delete/Erase Files, Compare Blocks, Copy Screens, Translate IBM Clusters, two versions of MSDOS, and more! \$15.00 + \$2 p&h.

David A. Hill, 1159 South Shore Drive #12, Holland, MI 49423, is publishing a newsletter for the users of AERCO's FD-68 Disk Controller. Published a periodically, the newsletter contains tips, program conversions for disk, and more. Send a SASE for more information.

Paul Holmgren, 5231 Wilton Wood Ct., Indianapolis, IN 46254, asks you to try it before you buy it. Where else can you get this kind of deal? I.S.T.U.G. is selling copies of our BBS program. \$16.00 gets the program and full directions to setup your own BBS. To sample it, call (317) 898-3903 [7,1,E:24 hrs]. Too many features to mention here. Runs 99.99% trouble free.

GLENN Technics, Box 2760, Anaheim, CA 92804, has AccuDraw for color art and tech drawings on the TS2068 for \$19.95. AccuDraw features scaled distance and angle readouts, 2 screens to draw on, elastic elements, 2 speed stick driven cursor, plus 32 other functions like undo, fill, text, rotate, 5x7 copy, full screen and origin crosshairs, fedge fill, and color select. It supports your 8x10 copy S/W with Aerco printer I/F.

John Deering, 136 Neverbreak Drive, Hendersonville, TN 37075, is offering mods to PRO/FILE 2068 for use with the Rotronics Wafadrive System. The price is \$9.95.

FORUM

SYNCWARE NEWS welcomes correspondence from its readers. Some editing of letters may be necessary for brevity or to enhance clarity.

Computerfest Update

INDIANAPOLIS, IN- Frank Davis, Producer of the Midwest T/S Computer Fest, announced that plans have been finalized for the 1987 edition of that successful event.

The 2nd Annual Midwest T/S Computer Fest will be held May 2-3, 1987 at the Holiday Inn North, 3850 DePauw Blvd., Indianapolis, IN 46268. Hotel reservations may be made by calling the Holiday Inn North directly at (317)872-9790 or through the Holiday Inn Toll Free Reservation Number, 1-800-HOLIDAY. If you mention that you are attending the Computer Fest, the Holiday Inn is offering a reduced commercial rate of \$52.00 a night, unlimited occupancy. This hotel is a Holiday Inn Holiday, which offers plenty of activities and excellent facilities for not only those attending the Fest, but their family members as well.

Registration info for the Computer Fest can be obtained by writing to Mr. Frank Davis, 513 E. Main St., Peru, IN 46970. Please mark all correspondence FEST INFO and include a self-addressed, stamped business envelope. Please DO NOT phone. Admission to this year's Fest will be \$4/person, \$7/family if pre-registered before March 30, 1987. Admission at the door will be \$6/person, \$9/family. Individuals and families may pre-register by writing to Mr. Davis at the above address.

The 1987 Committee has done a super job planning activities for everyone at this year's Fest. Beginning Friday night, a banquet will be held for the Vendors, participating User Group Officers, Speakers, and Committee members. This will be a social gathering with spouses encouraged to attend. The cost for the banquet is tentatively \$12 per person. Space is limited to 125 people. Advance reservations will be required for this event.

The Fest will run from 9am-6pm Saturday and 9am-5pm Sunday. Planned are a larger Vendor area, two concurrent program tracks with the most popular

***** SMART TEXT TS-2068 *****

*** A complete Administrative package: Word processor, Data base manager, Mail List, Full text editor, Automatic printer, Automatic formatter. Perfect letters or manuscripts in multiple copies. Mail List for personal form letters. Repeat print 5 page reports. Desk top publishing. Friendly, Menu Driven, no command keys to learn, Crash Proof.

*** Use with ALL Parallel Interfaces, Printers. Second program included for use with OS-64 Cart (64 Col Screen).

*** Disk Versions for Aerco FD and Oliger V2-12.

**** Satisfaction Guarantee. \$39.95 Check or M/O, Gulf Micro, 1317 Stratford Ave, Panama City, FL 32404. Specify type interface, for Cassette or Disk Drive, Inquiries welcome.

programs being repeated so all will have a chance to participate, and a separate area for a Timex Flea Market and Computer Swap.

Space in the Vendor area is being made available at three levels. Tables for retail vendors will be \$50/table in advance, \$65/table at the door. User groups may obtain a table in the vendor area for \$10/table in advance, \$15/table at the door. A new category this year will be Cottage Industry. This would be for the individual who has written a few programs and wishes to offer them for sale. These tables will be \$20/half table in advance, \$25/half table at the door. Space in the Flea Market/Computer Swap area will be free. It is limited to individuals selling used equipment and distributing public-domain software. Copying of programs other than public-domain software will result in the guilty parties being asked to leave. This rule will be strictly enforced throughout the Fest.

This year's Fest is shaping up to be THE grand event of 1987 in the T/S Community. People from both coasts and everywhere in between will be there. Don't miss out on the fun and the chance to meet and talk with the Vendors and other "famous" Timex- Sinclair personalities. Get those registrations in now!

The Fame & the Glory

The new SyncWare News programmers contest is rapidly coming to a close. All entries must be postmarked no later than March 15, 1987 to be eligible for the Educational Software Contest (see display ad inside for rules and details). To help you along with your own programs, our own Fred Nachbaur has consented to publish one of his educational programs. Part one of the two part "ABC/123" children's educational program appears this issue. This should whet your appetite and give you some ideas to get you going on your own programs for the big contest.

Big prizes await the winners, plus FAME and GLORY for all those who enter. Remember, you can't win if you don't enter. Get to it Programmers!

Who Is Orrin McGill?

Dear Editor,
Editor; that is a large job. I wish you luck in your new position. I will be monitoring your editorial style with curiosity.

To date, SWN does not offer bio's, addresses, etc. on the various contributors to the ZX81 saga. Yet one guy seems to have slipped by in the night. Who is Orrin McGill? He is marketing a product called 'PICTASCAN', what is it? There's also another product called 'MMRY-REZ 64K', what does it compare to?

We all know about Oliger, Nachbaur, and Kingsley. Are there any other ZX81 wizards out there? If so, they could probably use a little of the limelight too.

Carl Jones, Jr.
Medford, MA

Dear Carl, Thank you for your letter and good wishes. This is the kind of job that at times requires all the encouragement the readers can send my way. We'll try to keep up the fine standards established by my predecessors.

Orrin McGill is a Minnesotan who contracts to the E. Arthur Brown Co. as an independent software and hardware designer. He can be reached through his company, MIN-NY, at (612) 566-7094. The two products you mention in your letter are ones he developed.

MMRY-REZ 64K, alias ZXGR, is a 64K RAM and hi-res graphics pak for the ZX81. It will allow you to develop and store in memory up to five pages (screens) of hi-res graphics, design up to 256 sprites (or attributes), and includes software to allow an 'Etch-A-Sketch' function as well as program commands like 'circle', 'line', etc. This product is available through Mr. McGill or E. A. Brown for about \$99.00.

PICTASCAN, E. A. Brown, \$149.95, is a scanning device that connects to the printhead of an Epson FX-80, FX-85, or compatible printer. By using the printhead to scan back and forth, and the printer carriage to move the sheet through the printer, PICTASCAN will copy the graphics on a sheet of paper to the computer screen. It works in a fashion similar to 'Thunderscan' for the Mac computers. Currently, PICTASCAN is available for the IBM and compatible computers. Mr. McGill states that if there is enough demand, consideration will be given to a Timex version.

We appreciate your suggestion regarding giving more information about our contributors. Recently, because of some problems that occurred, it has been the policy of SWN to not publish an author's address or telephone number unless he or she requested us to do so. However, you're correct in stating that these folks deserve some of the limelight. As Editor, I'm going to commit myself (and probably be committed for doing so) to do two things for our readership. 1) Whenever possible, we will publish a short bio about the author, telling you a bit about their background and their system. 2) If you desire to contact a SWN author, you may do so through the Editorial Offices here in Ohio. Here's how. Send your letter to the author to me. Include an envelope with your return address and enough postage to mail your letter first class. I will forward your letter to the author. The author may then contact you if he/she so desires. We'll try this for a while to see how it works out.
-jm

External Power

Dear Editor,

In SWN 4:1, Barry Carter wrote that he was having problems with the video display when he had both his Zebra disk interface and a printer interface connected to his computer (a 2068). I had this problem with my Aerco disk interface and all I had hooked to the computer at the time was the disk interface. Your reply about the problem being the power supply was 100% correct.

As you probably know, when Timex built the 2068 they cut many corners so they could keep the cost down (and the profit up). One of the corners they cut was in the power supply. The power supply is just barely adequate to run the computer. Adding anything such as a disk interface that draws its power from the computer will start to overtax the power supply. In my case, the display was quite readable until the drives were actually accessed. Then the 12 volt supply buss dropped down to 10.5 volts. This low voltage condition was more than the modulator chip could tolerate. Its operation became erratic and I got a ghost image on my composite monitor. Even with a RGB monitor, the display would jitter a bit every

time the drives were used. Fortunately, I was able to rid my system of display problems by not powering the add-ons with the computer. I now supply the add-ons power via an external power supply.

To use a separate power supply, the first add-on plugged into the computer should have the 5 & 15 volt buss lines cut at the connector. The external supply should be wired in at this point on the add-on side of the cut. Connect the external power supply +5 volt lead to the trace that went to connector trace 4B, the +15 volt lead to the trace that went to 3B, and the GND or COM lead to 7B and 8B.

One point of warning is in order if you have wired in an external power supply to power your add-ons. The computer supply and the external supply should be plugged into the same power strip. This power strip should have an on/off switch so the computer and the external power supply can be turned on and off together. If you turn off the computer and not the external power supply, you could cause the add-on to operate. This could, say in the case of a disk drive, allow the disk drive to turn on and then operate in the write mode. Any data on the disk could be destroyed, even if the disk had been write protected. So remember, always have your system set up so that you turn the computer and external power supply on or off at the same time.

For those who would like to add an external power supply, there are several companies that sell power supplies that would be good for this purpose. One is B. G. Micro of Dallas, Texas. Check the ads in magazines like THE COMPUTER SHOPPER for others.

David Lockin
Los Angeles, CA

A Monitor Is...

Dear Editor,

I have a few questions regarding color monitors that I hope you can answer.

- 1) Is an RGB color monitor the same thing as a Hi-Resolution color monitor?
- 2) A lot of the monitors I've seen advertised mention Horizontal and Vertical resolution (ie., so many dots horizontally, so many vertically). What does this mean?
- 3) Can any or all color monitors be used with the TS2068 in its expanded 64 column mode or does the monitor have to be capable of handling 64 columns?

I am going to purchase a color monitor for my TS2068. I am hoping that your response will help me.

Dennis Silvestri
New Haven, Conn.

Dear Dennis, Your simple questions regarding monitors require very long and detailed answers. So long in fact, Fred Nachbaur, our technical editor, is currently writing an article comparing TV's, composite monitors, and RGB monitors and describing what to look for as far as resolution, etc. The article will hopefully be ready for printing in the very near future.

Here are a few rules of thumb for you concerning monitors until Fred's definitive article makes its debut. Unless you have added a circuit that you KNOW will drive a RGB or TTL input monitor, DO NOT BUY

ONE! Unless you have modified a ZX81 or TS1000, it will not work with a composite monitor. The TS2068 will work with either a TV or composite monitor. Most monitors manufactured today have a screen which is designed to handle a display 80 columns wide, 24-25 lines long. Such a composite monitor will be more than adequate to display the TS2068's 64 column wide expanded mode screen.

You might want to check out a few back issues of RADIO-ELECTRONICS and COMPUTERS & ELECTRONICS. They both have run articles on monitors (COMPUTERS & ELECTRONICS is no longer being published). The December 1984 issue of COMPUTERS & ELECTRONICS has several good articles on monitors.-jm

ZX81 Flight Trainer

Dear Editor,

I am an avid subscriber to SyncWare News. As one who has not given up on the Timex Computers, I have come up with an interesting application for a ZX81 computer. I would like to interface a ZX81 to a Pacer MK II Flight Procedures Trainer. The ZX81, its associated software, and an A to D converter would be used to display the horizon, show the runway upon approach, etc., based on the analog outputs of the trainer.

If you or any of your readers have ever performed such a task with this trainer or a similar unit, I would greatly appreciate hearing from you.

Carl Rink
807 Londonderry Dr.
Sunnyvale, CA 94087

Dear Carl, What a fascinating application!

My background is in industrial controls and real-time interface applications. Unfortunately, the work with SWN leaves me little time for outside projects. However I can recommend a book that might give you some insight into getting your little project to work. TIMEX/SINCLAIR INTERFACING by J. Downey and D. Rindsberg, and published by Prentice-Hall (c) 1983, has a wealth of information between its covers. I am also printing your address along with your letter in case any of our readers can help.

When you do get this project up and running, I'm sure that the rest of us would be more than interested to see how you accomplished the task. Your experiences in this hands on 'class room' will make excellent reading. Keep us informed of your progress.-jm

What's a MIDI?

Dear Editor,

I saw the article on the new Spectrum 128 in SWN 4:1. One of the machine's features was an onboard MIDI interface.

Since the MIDI first hit the market, I've searched for one that would connect to either my TS1000 or TS2068. Needless to say, I haven't had much luck. Presently, I see my options as to either purchase one and attempt a modification or to build one from scratch.

My question: What comprises the circuitry of the MIDI? Is it typical of other I/O devices or does it have specific differences? Most MIDI's I've seen

advertised are designed for the Apple, Commodore, or IBM. Is the Z80 capable of supporting this device or is it too antiquated?

Thank you for your assistance and all the support your fine publication has provided.

Daniel Wade
Oxon Hill, PA

Dear Daniel, There is no reason to re-invent the wheel on the MIDI in this case. The Z80 micro-processor is more than capable of handling just about anything you'd care to throw at it. Its only limitation is usually the programmer's ability. Our technical editor, Fred Nachbaur, informs us that the heart of a MIDI device is simply a D to A converter, and the way to a MIDI's heart is through its serial port. If you have provisions for a serial output port on your computer and obtain a MIDI with a serial input port, you should be able to make the hardware hookup. The software to drive the MIDI is going to be up to you, as there is not any commercial Timex/MIDI software on the market. Tim Woods, TIME DESIGNS Magazine, 29722 Hult Road, Colton, OR 97017, has, at different times, solicited for interested MIDI users to contact him. Perhaps he can offer some advice.

If you do come up with something interesting, drop us a line and let us know.--jm

We're only Human Dept.

Dear Editor,

By now, probably 1,066 readers have written to tell you of the error on page 4 of SWN 4:1. Tom Bent must have been thinking of a more powerful computer when he said to...

LET YES\$="SI"

As far as I know, the 2068 will accept only 26 simple, single-letter labels.

Basil Wentworth
Bloomington, IN

Dear Basil, Yes, we have received about 1,066 letters on this one. Here's letter number 1067.--jm

Dear Editor,

This is in response to the letter in SWN 4:1 from Sergio Frost of San Diego, CA.

A program that lends itself well to the task of computer translator is PRO/FILE by Tom Woods. This simple but fast file program is like a computer index card box. All you need to do is to add files like this:

YES=SI

This file would be found when searching for either Yes or Si. More complex files could be done as well. This application is mentioned in the manual that comes with the program. If you don't already have PRO/FILE, you should get it for all your filing needs.

Bret Lanius
Decatur, GA

Dear Bret, Thank you for your letter and recommendation for the PRO/FILE program. It would indeed fit the bill as a translator program. --jm

Move Over NASA!

Dear Editor,

I thought you and your readers might like to hear about an interesting TS computer application.

I've occasionally been asked what useful purpose my computer has; over and above what could be done without it. I used to kid, "Why I can catalog its software and hardware." Sometimes I'd be boringly practical, like "I use it for word processing, keeping track of finances or taxes, etc.". Well I finally dropped the mouth open of the last would be heckler. "I've used it to win several contests. In fact, I did so well I was accused of cheating."

One of my other hobbies is building and flying model rockets. I used a modified simulation program to predict flights to optimize the performance. In one contest I was twice as good as second place, hence the cheating accusation. I was accused of not using the computer (which everyone knew), but of using a rocket engine of a more powerful content than my entry class allowed. Fortunately, all rockets were pre-inspected and my honor remained untarnished. I actually took the unfounded accusation as a compliment; my efforts to do better worked! After several wins, the club members are now interested in learning how to use the computer to improve their flights. I plan to conduct a seminar at a coming club meeting.

To further the use of the computer in this area, I plan to develop a transportable 115Vac/12Vdc system complete with printer for next season's flights. I will have a 4-1/2" dc monitor, CAI stringy floppy, and a TTX 6 volt printer all mounted into a home made plywood case. I have visions of a real neat package capable of use in my car or motel room when I travel.

Mort Binstock
Pittsburg, PA

Dear Mort, Congratulations for once again dispelling the myth that our computers are 'toys'. It would also seem that you have been successful in merging two hi-tech technologies for hobbyists. If sometime in the future you would care to publish your rocket programs, we would be more than happy to print them in this magazine.--jm

Pat's Soapbox Just for the Fun of it

by Patricia Kardokas

Why are you still wasting time with that toy? Why don't you get a Mac, an IBM, or one of the clones? We have all heard these questions many times. Don't you get tired of it? Of course you do, and so do I. Let me give you a few of the responses I give to people who down grade our TS computers.

The Kardokas Family computer is a TS1000 with a Byte-Back UM-64K Ram Pak and a Memotech full-size keyboard. The printer is a TS2040. In addition there is a Zebra joystick interface. We have a complete assortment of software ranging from entertainment to serious applications. The software was obtained from a number of sources, and includes both original Timex titles and packages from third party sources.

One of my favorite pieces of software is a word processing program called 'Fast Word' which was published third party by International Publishing & Software Co. IPS published quite a bit of software including 'Packrabbitt' and 'Krakit'. For those of us who have only TS2040 printers and do not want to invest in printer interfaces and 80 column printers, 'Fast Word' is a great word processor. Beside the 'normal' features found in word processor programs, 'Fast Word' has the ability to produce printouts that run sideways on the paper. By taping the strips of thermal paper together and running them through a photo copying machine, a normal 80 column page can be produced (I've seen the results. It may be a little extra work, but the final product looks ok.-ed.).

Our system's mass storage device at present is still a cassette tape deck. Planned additions include a floppy drive and the Olliger Video Upgrade. The floppy disk drive is not an urgent matter because we have not had any serious problems with cassette loading and saving. Our system is completely reliable. A few minor problems have occurred with the various interfaces stacked behind each other in a string. They do occasionally loosen up, causing some strange things to happen, but that is part of the adventure of TS computing.

Every new computer which comes out has a superfast processor running it and amazing amounts of RAM. I have a home computer system. I am not running a major corporation. Home computerists don't need that amount of speed or a half or full megabyte of RAM. In our world, 64K is enough for our needs.

Will our family keep on using our Timex computer? Sure we will! Will we continue to expand and enlarge the system? Naturally we will. As long as the people and companies we hear about in publications such as 'SyncWare News' and 'Computer Updates' continue to provide us with a way to compute with our small systems. We are doing things with our systems that

people tell us can't be done. They say our machines are too small for serious computing. What is serious computing? Serious computing is whatever the user wants it to be. With companies like IBM and Apple, the manufacturer tells you what your limits are. With our systems, we define our own limits. Our only limit to what can be done is our own imagination. To slightly paraphrase the old Timex slogan, "THE POWER IS IN YOUR MIND." If it can be done, one of our users will do it. Once they're successful, another small TS business will be born. If the product proves to be good, the Kardokas family will buy and enjoy it. Computing is fun. It is an adventure.

To boil it all down, the reason that we use our Timex computers is that they are fun to use. They are easy to use. The hardware add-ons and the software are inexpensive. The software and hardware has been designed to get the most out of our small systems, especially since the crash of the Timex Computer Corporation. It allows us to do things that other people are spending huge sums of money to do. When you are limited to an absolute maximum of 64K of usable RAM, you have to be more efficient. You can't be as wasteful with your resources. We have seen that great programming and hardware design can overcome all the problems caused by the small size of our systems. Our systems are as good as any other home computer in spite of their size.

Let's keep doing it gang. Let's keep the fun in computing!



Patricia Kardokas is married and the mother of two children. Patricia, her husband, children, and TS Computer reside in Worcester, Massachusetts.

EXTENSIONS TO PRO/FILE REVIEW

LEE BENNETT

There are four issues of EXTENSIONS that total 31 pages of well documented changes, additions, and instructions. These modifications can either be keyed into your computer or you can purchase the programs on tape. The tape, which includes issues 1-3 and Thomas Woods' BREAKTHROUGH code, will allow you to merge the first three issues to your existing PRO/FILE program in just minutes. Issue #4 will have to be keyed in manually irregardless of how the first three issues are done.

EXTENSIONS requires that Thomas Woods' BREAKTHROUGH modifications be made. If you purchase the printed issues of EXTENSIONS, you will need to key in BREAKTHROUGH also.

EXTENSIONS and BREAKTHROUGH can be obtained as follows:

Robert C. Fischer
Rt. 2 Arizona St.
Emerson, Ga. 30137
EXTENSIONS issues 1-4/ \$6.00 each
EXTENSIONS tape, contains issues 1-3, plus
BREAKTHROUGH/ \$18.00

Thomas Woods
P.O. BOX 64
Jefferson, NH 03583
BREAKTHROUGH, a newsletter not a tape/ \$7.95.

I have all four issues of EXTENSIONS and the tape. I have keyed in 90% of the modifications myself, plus I have merged the EXTENSIONS tape with my own tape to see what would happen. My advice is to read and reread the instructions. Do exactly what they say to do. My short cuts did not work and they cost me several long distance telephone calls to find out that I should learn to follow instructions. Mr. Fischer is a high school teacher and knows how to write instructions. Read all of the instructions, then reread them before beginning the modifications. This will save you time, money, and frustration.

All the routines work as they should. All are useful, but maybe not for everyone. If you are an above average programmer, then you may be able to input only the specific modifications you need and save memory and time. The serious PRO/FILE user will

probably want all the routines available. Some of the more commonly used modifications are described below. From this point on, I will refer to the PRO/FILE program with the EXTENSIONS modifications as PLUS 4.

The new SAVE FILE routine allows you to save the program and file separately. This can save time when loading and saving short files using tape. You can also append one file to the end of another file as long as their combined length does not exceed the limits of D\$. After you append files, you can perform all the old and new actions on them. As a side note, D\$ has been reduced to 27500 bytes. This was necessary to gain memory space for the additional routines contained in EXTENSIONS.

There are two Update routines:

Update #1 works like this; Type a "Search Command" | U. The program will ask you, "What line do you wish to change?", "What data will be changed?", then "What is the new data?" After answering each question, hit "ENTER". The program will begin to automatically search and replace data in each file. This will only operate on phrases or blank spaces that begin with the left margin. Update #2 works differently. Type in the command "EDIT". The program prompts, "EDIT PHRASE". You type in your phrase. The program returns to the main menu. Nothing happens. When you need the "EDIT PHRASE" in a file, place the cursor anywhere on the screen where you want the "EDIT PHRASE" to appear and hit "AT". If you are in the "OVER MODE", it will print over what is already there. If you are in the "INSERT MODE", the phrase will be inserted.

EXTENSIONS also makes it possible to reverse a two word phrase, such as a name, and print the first word last. You simply type the two word phrase as you want it alphabetized. In the case of a person's name, this is usually the last name first. When you want the reversal routine to work, activate the "AUTO" mode. Before this feature became available, I put my names on two different file lines. Line one had the last name first then first name with no space between. This was for the purpose of alphabetizing. On the second line, I had the name as I wanted it printed on a mailing label. I sorted by the first line and printed out the second line. Now I have an additional line to use for data, because I can erase the second line and put a space between the last and first name. The program needs a space to know where to locate the second word. Only the first two words on the first line of the file are involved. Anything beyond the second position will be ignored.

With EXTENSIONS, you no longer have to go to the main menu to ADD or Edit a file. You can go directly to the next file by using some new commands. (See FIG. 1)

TALLY and SUM are two different mathematics routines. Here is what Sum will do:

SUM = arithmetic on a whole file line.
SUM 1 = Running a total of all files.
SUM 2 = Total of just the file on the screen.

Here is what TALLY will do:

SUM 1 = Total of left side of screen.
SUM 2 = Total of right side of screen.
Both SUM and TALLY will do arithmetic, such as;
50+20-10.

Putting "##" in the first spaces on the first line of a file will prevent that file from being sent to the printer. I use this to keep my directory from being printed out.

When Mr. Woods wrote BREAKTHROUGH, I sent for it right away. I keyed in the machine code, transferred

1) From Main Menu:
EDIT:EDIT PHRASE (Update 2)
T: Key Tick
DATE: Set Date
Search Command | U:Update 1
Search Command | T:Tally
Search Command | S:Sum

2) From ADD or Edit Mode:
Step:Edit to Edit
TO:Add to Add
TO:Edit to Add
Copy & hold ENTER:Name reversal
THEN:Print from Edit file
AT:Insert EDIT PHRASE
AND:Duplicate file

3) Misc:
##:Do not send to printer.
> :Middle Search, resume search.
>=:Middle Search to Print out, resume AUTOsearch to printout.
ENTER/SPACE:On/OFF-Name reversal in AUTOsearch mode.

Other New Improvements:

- 1) "ALL" defp command prints blank line between files
- 2) Auto caps
- 3) Make back up copies of program.
- 4) Will not print out "Search is Complete"
- 5) New way of avoiding print out blank file lines
- 6) Cursor wrap around
- 7) Sinclair microdrive set up for saving, loading and appending
- 8) plus more modifications to the program.

FIG. 1: SUMMARY OF NEW PRO/FILE PLUS 4 COMMANDS
(Robert C. Fischer's EXTENSIONS)

my files and started alphabetizing them. Unfortunately, my computer locked up on some of the files. I corresponded with Mr. Woods to find out why. He had no solution to my problem. Some time later, I received a telephone call from Mr. Fischer. He was experiencing the same problem. He wanted me to send him my program and files. Using his and my files, he found the why of the problem. Talking with Mr. Woods, it was decided the solution was a rewrite of the BREAKTHROUGH machine code. This was more work than either of them wanted to undertake. Mr. Fischer has an alternate solution to the problem. Since using his suggestion, I have not had a crash or locked computer. If you have had a similar problem, get EXTENSIONS #3 for a discussion of the problem and a remedy.

Before giving my evaluation of the modifications, I would like to make the following observations:

1. To accommodate the modifications, Mr. Fischer had to make room for them by assigning many variables. It takes lots of time and effort to redefine the variables, as well as enter all the new routines. There are lots of places you can make mistakes. If you are not good at programming, then order the tape.

2. If you purchase the printed EXTENSIONS and intend to make the changes yourself, then realize there are changes to the changes. Between issues, Mr. Fischer continued his work and some times found new or better routines. So he made changes to the changes, when he felt it was necessary.

3. In the ADD and EDIT mode, the input seems somewhat slower as a result of the changes. If you are an accomplished touch typist, then you may get frustrated at the slower response. It does not bother me, because I am an accomplished hunt and peck typist. (See note.)

4. Mr. Fischer has made no effort to include many of the new commands in the main or other menus. He has added no "HELP" page. I tried to add such a page, but found there was not enough memory left without reducing D\$ even more.

5. Depending on the way you purchase PRO/FILE, BREAKTHROUGH, and EXTENSIONS, the total cost runs anywhere from around \$53.00 to \$62.00 (the tape versions are actually more economical). This sounds very expensive. Then I remembered, I paid \$69.00 for MSCRIPT. I added two modifications that upped my total cost for MSCRIPT to \$82.00. Still, for good business software, it is worth the cost. Remember, you do not have to purchase all the issues at one time. You can spread the cost over several months by purchasing one issue at a time. As you complete an issue, you can use the modified program while you are adding the next issue to a separate copy. The files are saved separately and can be loaded into the revised programs.

In my opinion, six items need to be corrected to improve EXTENSIONS:

1) The literary style of the printed version changes from the first issue to the last. In issue #1, Mr. Fischer used headings for each section, but by issue #4 there were no headings. All the information is there, but headings would make it easier to find specific topics. Headings are handy if you forget how the routine is activated and you want to look back into the documentation to find the instructions.

2) The changes to the changes should be fixed.

3) I believe EXTENSIONS should be reworked to make all four parts into one whole. Having all four issues on one tape would eliminate the need to manually key any of the modifications into your computer.

4) I do not like the reduced file size even if it is just 500 bytes.

5) I do think a "HELP" page would improve user friendliness.

6) The total cost of just the EXTENSIONS programs could be as much as \$31.95. This is more than the original PRO/FILE cost me. If you do not need the printed EXTENSIONS, I would just order the tape and issue #4. This will keep the cost down some.

I would recommend PRO/FILE plus EXTENSIONS (PLUS 4) to anyone who wants a good data base that has been made even better. Mr. Fischer is friendly and helpful. Give him a call if you have problems. He is fairly fast depending on his work load at school. In most cases, you will have your tape in 7 to 10 days. I don't know if he holds orders until checks clear. You might want to telephone him and ask, if delivery time is important. Mr. Fischer has made PRO/FILE even more flexible and useful. I have MASTERFILE, but I think PRO/FILE plus EXTENSIONS is better for my needs. If you use PRO/FILE, you need EXTENSIONS.

Note: As a courtesy, I mailed Mr. Fischer a copy of the first draft. His only comment was regarding observation #3. He says key response is not slower. You be the judge.

TYD☆BYTS

Another Crash Fix

by Mort Binstock

Several suggestions have been made in the past to prevent the mysterious crashes in our computers; using Z80B CPU's, 5vdc regulator cap, flexible cable to the RAM, etc. to name a few. Here's another.

One of my units became intermittently bump sensitive. The problem turned out to be a defective socket. I located the problem by carefully probing each chip pin with a straight pin. The fix was a shortened straight pin jammed between the chip and the socket.

The SyncWare News

Thomas B. Woods
Jeffrey D. Moore
Thomas J. Bent
Fred N. Nachbaur

Publisher
Editor
Associate Editor
Technical Director

All subscription information, billing and ad artwork should be addressed to:

SyncWare News
PO Box 64
Jefferson, NH 03583

Phone: (603) 586-7734

Article submissions, Forum, Support listings, Classified ads and any questions you may have concerning the use or everyday abuse of Timex Sinclair computers may be sent to any of the editorial offices.

Jeff Moore
602 S. Mill St.
Louisville, OH 44641
(216) 875-1257

Tom Bent
9016 Flicker Pl.
Columbia, MD 21045
(301) 730-7187

In Canada

Fred Nachbaur
Mountain Station Group Box C-12
Nelson, BC V1L 5P1 Canada
Phone: (604) 354-3858

Back issues of SWN are available for \$3.50 each. Volume #1 (for the TS1000) is available for \$16.95

SWN is done entirely on Timex Sinclair computers. Submissions are preferred as word processed text files (so we need not reinvent your article) on good tape. Memotext for the 1000, Mscript for the 2068, QLWP for the QL are the preferred word processors. We can also handle RCPM files done in WordStar (or Ascii format CPM files). Tasword files must be done in a 52 column format. Programs must be submitted on tape, so that we can verify that they will run. Label the tape itself, in case the documentation gets separated from the tape. If you do not have a word processor and have a good idea, then don't hesitate to write up your thoughts and submit them. SWN pays about \$40 per page for original articles. This amount varies slightly from issue to issue and is paid when the article is published. Write for more information.

announcing The SyncWare News 2nd (sort of) Annual Programmers' COMPETITION

Syncware News is sponsoring another Programmer's Contest. This time around, we're looking for the best in interactive educational software for children. The winners and the best of the rest will be considered for publication.

Interactive educational software, in this case, is not limited to Addition or Spelling games for first graders, although these are surely welcome. We'll also consider educational simulations, demos, and the like. Artificial Intelligence can come into play here, too. A simulation program that would let a student see how a saw mill works, for instance, is an educational program. It becomes interactive when it asks questions about what the child has seen demonstrated and rewards for the appropriate response.

Prizes!

GRAND PRIZE - Best Program Over All
Writeman NLQ Dot Matrix Printer - features include 160 cps; 8K buffer; Epson compatible print commands; 9 wire print head; Standard, Italic, and Proportional print characters; down-loadable graphic characters; Retail Value, \$400.00

FIRST RUNNER UP - TS2068 DIVISION
32K Non-Volatile RAM Cartridge Board - features include 32K of non-volatile RAM which can be mapped to reside in either the Cartridge Bank or the EXROM Bank; Battery Backup to prevent data loss; runs from 100% battery power or will draw power from the 2068 while the 2068 is powered up; Write protect switch; Easily modified Chunk specifications; 32K NVM can be used to mask EXROM or contain LROS and AROS programs; Retail Value, \$110.00

FIRST RUNNER UP - TS1000/TS1500 DIVISION
Byte-Back MD-2B Modem - features include the ability to upload and download text to and from tape and memory; 300 baud - word length, parity, etc. selectable; direct connects to telephone line; RS-232 port; Retail Value, \$129.00

Rules

Deadline for entrys is March 15, 1987. The winners will be announced in the July/August 1987 issue of SyncWare News. All entries must include a CASSETTE TAPE of the program(s), and should be mailed to:

SOFTWARE COMPETITION
c/o SyncWare News
P. O. Box 64
Jefferson, NH 03583
ATTEN: CONTEST

Entries will be judged on the basis of:

- 1) MEETS CONTEST CRITERIA - is it educational?
- 2) DOCUMENTATION
- 3) EASILY MODIFIED - can I easily insert my child's spelling words into a spelling program?
- 4) AESTHETICS
- 5) EFFICIENCY and SPEED OF CODING
- 6) TS2040 PRINTER SUPPORT (but not mandatory)

All entries must work with NO ADDITIONAL HARDWARE (except the TS2040 printer, and 16K-64K RAM). Entries may not make use of additional commercial or proprietary software. This will make judging entries less of a monumental nightmare, although support of other hardware or software options will be taken into consideration. The finalists will be judged by a panel consisting of the staff of SyncWare News. All decisions of the judging panel are final.

All entries become the non-exclusive property of SyncWare News. SyncWare News reserves the right to publish winning entries in upcoming issues. EVERYONE is eligible to enter except the staff of SyncWare News. East coast entries will be judged on the West coast and all western entries will be judged in the East. We are presently soliciting user groups to judge entries.

Remember, you can't win if you don't enter. Get started right away!

Super Prizes, Fame & Glory can be YOURS!

RUN YOUR MEMOTECH 64K ON THE 2068

Kent E. Cook

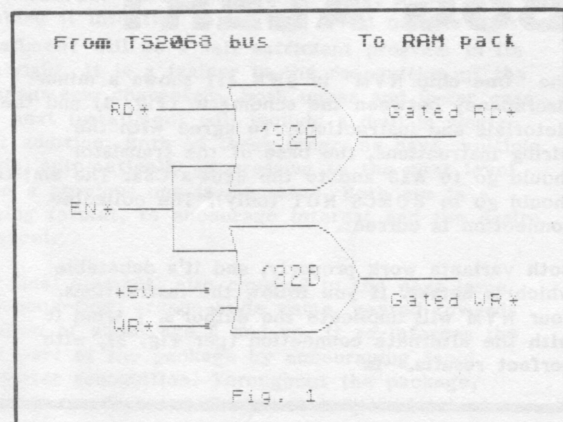
I am currently building a 4464 DRAM board for my continually evolving ZX-81 based system which I have on S-100 boards. This will leave me without a use for my Memotech Memopak 64K. Feeling that \$179.95 was a bit much for a paperweight, I decided to attempt to add it to my 2068 system in a manner which would not require additions or modifications to the RAM pack.

I added bank switching features to my ZX-81 system a few years ago which allow me to switch between RAM, EPROM, peripherals, and memory-mapped I/O. My first attempt at implementing this was to gate the MREQ* signal to the RAM so that the RAM did not receive MREQ* when a selected memory-mapped device was to be enabled. This failed completely, and, without RAM for the operating system, was impossible for me to troubleshoot. I attempted to get advice from Memotech and discovered their technical data is vital to Great Britain's national defense and I needed approval from Mrs. Thatcher or an act of Parliament.

Next I tried gating the RD* and WR* signals. This worked well with the Memopak 64K. I have the published schematics for the John L. Olliger 64K Memory Board and the Gladstone 64K Ramkit and I see no reason why these should not work too. (It should work with any brand of 64K ZX-81 / TS1000 compatible RAM - However, I can give no guarantees.)

The theory and hardware implementation are both simple. For a RAM to output data to the bus it must receive, among other things, a RD* from the bus master. Likewise, to store data, the RAM must receive a WR* signal. If neither of these lines are in their active low state, the RAM will simply sit passively on the bus structure, unaffected by its surroundings (except for refresh, which is a horse of a different color).

If we insert the circuit of Fig. 1 between the TS 2068 bus and the RAM pack we make the RAM's receipt of RD* and WR* conditional on the state of EN*. If EN* is low the OR gates' outputs follow the inputs. If EN* is high the OR gates' outputs are held high, disabling the RAM. The optional switch in the input of the lower gate gives a write-protect capability.



If the input to the gate is held to +5V, the WR* signal is not passed to the RAM pack.

The EN* signal can be picked off the TS 2068 bus from pin 28A, ROSCS*, to give a 64K page of RAM in the Dock bank, or from pin 27A, EXROM*, for an extra 56K of RAM in the EXROM bank, providing the lower 8K of the RAM pack is disabled and you have fully decoded the TS 2068 EXROM (see SWN 2/3, pp19).

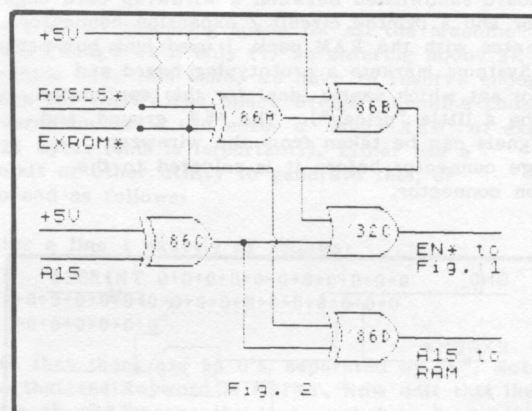
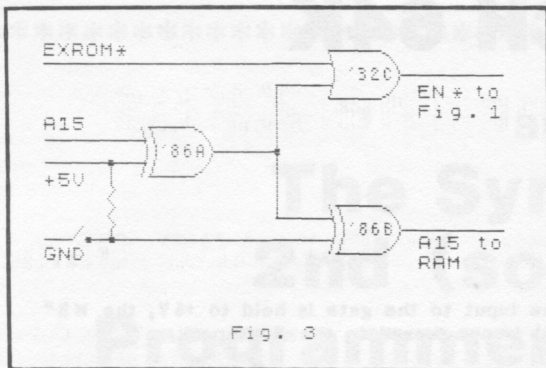


Fig. 2 shows one of many possible applications. (Was it Archimedes who said, "Give me an unlimited budget for hardware and software and a place to stand and I will move the world", or was it Fred Nachbaur?). If either ROSCS* or EXROM* go low, the output of 74XX86 gate A goes high. This state is inverted by '86 B and fed to the input of an OR gate. A15 from the TS 2068 bus is inverted and fed to the other input of the OR gate. This generates an EN* if the computer is looking at the upper 32K of either the EXROM or Dock bank. The A15 line to the RAM pack is passed through an XOR gate which inverts it if EXROM* is high. This decoding scheme yields 32K in the top half of both the EXROM and dock banks if your RAM pack allows you to enable the entire 64K. The optional 4.7K to 10K pullup resistors and switches permit disabling RAM in either bank.

In the circuit of Fig. 3, EN* is derived from EXROM* and A15 such that the RAM pack is enabled in the upper 32K of the EXROM bank. The switch and 4.7K to 10K pullup resistor determine whether A15 is passed to the RAM pack in an inverted or true state, resulting in two selectable 32K pages. This is a nice complement to Hot Z AROS in the dock bank. You could, of course, use ROSCS* instead of EXROM*.

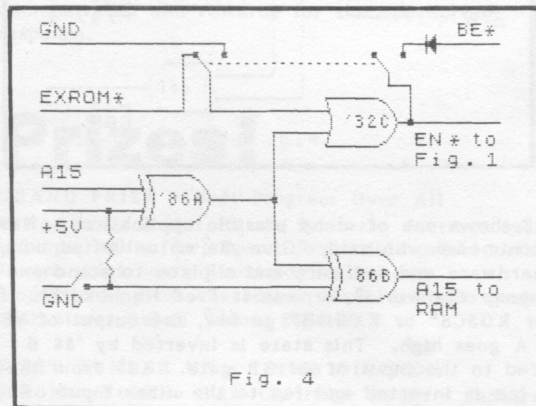
The circuit of Fig. 4 is the same as the previous circuit with the addition of a DPDT switch and a 1N4148 diode. With the switch to the left, as shown, the circuit works as before. With the switch to the right, the RAM pack is enabled for any address in the top 32K. The EN* is diode-ORed with the Bank Enable (BE*) input to the TS 2068 which causes the selected 32K page to be read by the TS 2068 regardless of the state of the horizontal select and chunk select



ports. This means you can SAVE and LOAD the RAM pack directly using mass storage devices which normally operate only in the home bank.

I put these circuits together using wirewrap sockets in perfboard sandwiched between a wirewrap card-edge connector and a printed circuit / expansion connector which mates with the RAM pack. I used junk box parts.

Zebra Systems markets a prototyping board and connector set which seems ideal for this application (if maybe a little "pricey"). The +5V, ground, and other signals can be taken from the wirewrap card-edge connector before it is soldered to the expansion connector.



There is some "fine print" to be considered. The RD* and WR* lines from the TS 2068 must obviously not be connected to the RAM pack through the expansion connector. The outputs of the appropriate gates are fed to the expansion connector to supply these signals to the RAM pack. The same applies to A15 if you use one of the circuits involving it. Do not connect the TS 2068 bus pin A5 (DZ IN) or pin B26 (DZ OUT) to the expansion connector. These lines are used by ZX-81 / TS1000 RAMs as RAM CS* (held high to disable internal RAM) and ROM CS*, respectively. If these RAM pack output signals are fed back into the TS 2068 "TS 1000 COMPATABLE" bus they will interfere with peripheral devices which use the daisy chain lines (JLO disk system, for one).

The JLO RAM uses the 5 volt supply line on the bus and therefore should work directly. The Gladstone RAMKIT and Memotech Memopak 64K have their own internal 7805 regulators which were powered by the ZX-81 / TS1000 "+9V" bus line. This input to the RAM packs can be powered by jumpering the TS 2068 +15V supply (pin B3) to the RAM pack "+9V" input (between the +5V line and the slot). Do not connect the TS 2068 bus pin B5 to the expansion connector thereby keeping the +15V supply off the "NC" B5 line.

If you are attempting these ideas with another brand of RAM you must be sure it is safe to apply +15V to the nominal +9V line to the RAM. I once fried a 7660 voltage converter in a Memotech RS-232 i/f with 16V. If you can't determine how your RAM works, you might consider pre-regulating the TS 2068 +15V using a 3-terminal regulator IC on the board with the gates.

One possible alternative would be using a ZX-81 / TS1000 +9V DC supply connected to a jack on the perfboard. Connect the power supply ground lead (ring) to the bus ground line and the +9V lead (tip) to the RAM pack +9V line. Again, as a safety measure, do not allow this to get back to the TS 2068.

As the gating interferes with RD* and WR*, the RAM pack must be the last device in line. I have mine at the end of a trail of JLO disk system boards, printer port, and EPROM burner and it all works fine (so far)!

All the circuits using EXROM* and ROSCS* depend on the state of the TS 2068 horizontal and chunk ports. Bank switching in BASIC and machine code has been addressed (pun intended) in past issues of SWN. Remember, home bank chunk 0 has the interrupt handler and chunk 3 normally has the machine stack. Expect problems if you switch these out without proper preparation.

I implemented these circuits with 74HCTXX devices. 74LSXX and 74ALSXX should work just as well. It is possible that 74CXX and 74HCXX devices would work, but CMOS inputs driven by TTL outputs are "iffy". All uncommitted inputs should be connected to +5V (a must for standard CMOS devices, and good practice for the TTL families.)

These circuits are "no frills". If I were designing them for marketing I would add filtering/bypass capacitors to the power supply lines, avoid mechanical switches in the data paths, use Tri-state or open collector logic for BE*, and maybe add some LEDs to indicate which page was active and show write-protect status. However, they work "as is (as are?)", are inexpensive and practical. I hope you can use them to advantage.

BUG ALERT

The "One-chip NVM" in SWN 4:1 shows a minor discrepancy between the schematic (Fig. 3) and the pictorials and instructions. To agree with the wiring instructions, the base of the transistor should go to A13 and to the 6264's CS2. The emitter should go to ROMCS NOT (only). The collector connection is correct.

Both variants work properly, and it's debatable which is better. If you follow the instructions, your NVM will duplicate the author's. I tried it with the alternate connection (per Fig. 3), with perfect results. -fn

ABC—123

by Fred Nachbaur

(EDITORS NOTE: Thanks to the generosity of Mr. Nachbaur, we are able to provide our readers with this two-part article as an example of how an educational program can be designed and one approach to the educational goals these programs hope to attain. It is our sincere desire that you type this program in and study it in order to get the creative juices flowing for your own offerings to our Educational Program Contest-jm)

Introduction

Our children are coming into a world in which computers are a fact of everyday life. Just as we grew up taking radio, television and automobiles for granted, so the youngsters of today perceive computers as commonplace items. Unlike many adults, who still find computers complex, and slightly menacing, our kids see them without bias or prejudice.

Most children are fascinated by computers. They will watch with interest while you use your computer, even if they don't know the meaning of all those things on the screen. They would love to participate, but the programs used by adults are simply not suited to their level.

With this in mind, we decided to run a SWN CONTEST for educational software. To get the ball rolling, we present a program that caters to young children, starting at about age 3. Interestingly, the program contains several routines from "CLEAR THAT SCREEN" (SWN #2:4), which we used to kick off our display contest. It is also an application of G. C. Harder's "SW LO-RES" (SWN #3:1) which was the TS1000 grand prize winner of this previous contest. The circle has thus gone around, from utilities and tools to actual applications.

This program is also an example of the "dedicated software" mentioned in a recent editorial. Have you upgraded to a more sophisticated machine? If so, use your "old" ZX81/TS1000 to give your kids, or your friends' kids, a head start in the computer world.

The "ABC/123" program is quite long, so we are breaking it into two parts. The result of this first installment will be a self-sufficient program in its own right. It is a trainer in the recognition of the alphanumeric characters, both upper and lower case. The next installment will include a drill in single-digit addition. Both of these programs have multiple levels, automatically progressing to the next level when a previous one is mastered. Both are in a gaming format, to encourage interest and the desire to excel.

The final element, also covered in the next issue, is a bona-fide, arcade-style game, geared towards children of about age 5 and up. It re-inforces the first part of the package by encouraging rapid character recognition. Throughout the package, graphics are fast and flashy, to help maintain interest.

Part 1: The ABC Game

In this installment, we will be entering approximately one-half of the ABC/123 package. Included will be the SW-LO-RES core, various machine-code screen utilities, the main menu, all the BASIC subroutines, and the "ABC" portion of the program (menu options 1 through 3).

The Machine Code

Start off by reserving space for all the machine-code. Though we'll only (!) be entering about 2K of machine code this time, we have to reserve enough space for the "Alpha Space Bridge" machine-code covered in Part 2. So enter a "mega-REM" of exactly 5021 bytes (this is important!). Either use a toolkit or other utility to generate this, or proceed as follows:

Enter a line 1 exactly as follows:

```
      1 PRINT 0+0+0+0+0+0+0+0+0+0+0  
+0+0+0+0+0+0+0+0+0+0+0+0+0+0+0+0  
+0+0+0+0+0+0
```

Note that there are 33 0's, separated by "+". Note also that the keyword is PRINT. Now edit this line (shift 1), and change the line number to 2. Continue doing this, until you have 18 identical lines, numbered 1 through 18. Finally, enter line 19 exactly as follows:

19 PRINT ,,,,0+0+0+0+0+0+0+0+0
+0+0+0+0+0+0+0+0+0+0+0+0+0+0

Note that line 19 only has 22 zeros. Verify that you have the correct length by entering PRINT PEEK 21535 and PRINT PEEK 21536; both should return 118. Then enter the following POKES (no line numbers):

POKE 16510,0
POKE 16511,159
POKE 16512,19
POKE 16513,234

There. You've just made a 5021-byte "0 REM" statement. Now enter line 1 GOTO 9000 and NEVER delete it!

Add the lines of the hex loader program. After proofing it, save to tape for good measure. Then, take a deep breath and start entering the hexcode of Table 1. Each section represents one full screen of hex strings. RUN 7000 to start entering each screen. First enter the START address given at the top of the section. Then enter each line of hexcode (32 characters), a line at a time. After entering the last line of the section, the computer will print the checksum for that screen. The number should match the checksum given at the end of the section. If it doesn't, you have an error somewhere. Find which line(s) are incorrect (numbered 1 through 20), and enter that line number, followed by the corrected line. When the checksum matches, enter "Y" to the "OK?" prompt. The program will stop, to allow you to save to tape, or go on with the next section. Take your time, save to tape frequently. (If you get impatient, you'll be sorry!)

1 REM "MEGA-REM",5021 BYTES

```
7000 REM "MEGA-REM"
7010 FAST
7020 CLS
7030 PRINT "START=";
7040 INPUT BASE
7050 LET SUM=0
7060 FOR L=1 TO 20
7070 GOSUB 7300
7075 IF AD>21535 THEN GOTO 7090
7080 NEXT L
7090 PRINT AT 21,0;"-51-3-";5
UM;"
7100 INPUT Y$
7110 IF Y$="N" THEN GOTO 7200
7120 IF BASE<>18434 THEN PRINT A
T 21,0;" RUN 7000 TO INPUT NEXT
SCREEN"
7130 STOP
7200 PRINT AT 21,0;"WHICH LINE I
S WRONG (1-20)?"
7210 INPUT L
7220 GOSUB 7300
7230 LET SUM=0
7240 FOR N=BASE TO BASE+319
7250 LET SUM=SUM+PEEK N
7260 NEXT N
7270 GOTO 7090
7300 INPUT L$
7310 PRINT AT L-1,0;L$
7320 FOR N=0 TO 15
7330 LET V=16*(CODE L$(2*N+1)-28
)+(CODE L$(2*N+2)-28)
7340 LET AD=BASE+16*(L-1)+N
7345 POKE AD,V
7350 LET SUM=SUM+V
7360 NEXT N
7370 RETURN
```

The Basic

When you're done with all seven screens, you will have entered the entire SW-LO-RES core, complete upper & lower-case bit patterns, and the various screen display utilities. The next step is to enter the BASIC. Note that lines 1 through 10 must be EXACTLY as shown! Entering PRINT PEEK 21609 should return 128 (the inverse square at the end of line 5). Don't run the program until you've entered all the lines, and have proofed them well.

When all is well, enter RUN 9990 to save to tape. When reloading later, the program will auto-run.

Using the Program

The first thing that comes up is the "Parent/Teacher Section," which prompts you to enter the pupil's name. Henceforth, the program will "remember" it, and it won't be long before your child recognizes it as his name. Since only eight large characters can be displayed across the screen, limit your entry to 8 characters.

You are now taken to the menu. At this point, only the first three options will operate. The rest will tell you, "Not installed yet." Pressing a key from 1 - 7 selects the corresponding drill or game. With some options, additional information is required. Going on, pressing any key except BREAK will flash the message, "HELLO, (name)" in big letters. Throughout the program, this flashing is a signal to the pupil to press "ENTER."

TABLE 1: HEX-CODE FOR ABC/123, PART 1
START = 16514

```
76760087048302850681018605820384
07802A0C4023E501840009019402D1ED
B00E84ED4218122A10402BE5018400A7
ED42019402D1EDB823E53E04011F0070
545D13EDB02323D20F2E1220E400101
280B78B120FBC9AF320940ED4B164003
03C5AF3221400AFE762009C12A0C4023
220E40C9FEDB2006CD9440C118E2FEDC
2005CDA94018F4FEDD2019030AFE21CC
EE40280DD61C3804FE043802CF153209
40E118BCFED8202E2A0E401122007EFE
76231B20F93A0940BA2804FE0220041E
63180D7BFE053002CF14D6045F2A0C40
19220E4018A5A717173002CF1BCB4728
05322140CB87171600CB125F2100433A
0940BD19282A113C40D5F50E010608E5
B9C57E170D20FCCB1B23C110F47BD112
130C3E09B9E120E5F13D2803E118DAE1
0E04060456235E23E5AFCB1217CB1217
CB1317CB1317218440856F7E2A0E40F5
```

CHECKSUM = 28400

START = 16834

```
C5E57EFE762002CF1AED4B1040A7ED42
3802CF0AE1C13A2140FE002805F1EE80
1801F17723220E4010BFD5111D001922
0E40D1E10D3801713DBA280BF5AFD516
B92804FE02200719220E40C3FD40ED4B
1040E5ED42E138F31EDF18EB00000100
101183002A0C40AFED521121007880F5
78FE0D300171F13D28032318F23E1190
5778C610FE1D3005FE053801713DBA28
07D5160019D118EC7880F578FE0E3001
71F13D28032B18F23E11905778C610FE
1E3005FE063801713DBA280BF5AFD516
00ED52D1F118E811220019109DC92A0C
400618237EFE76200310F8C9C6807718
F2002A0C4023E511210019D101B502ED
B0D511210019012000D1EDB0010015CD
F508C900000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
```

CHECKSUM = 24638

START = 17154

```
0000000000000000F0F0F0F000000000F0F
0F0F00000000FFFFFFF0000000000000
0000F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F
0F0FF0F0F0F0FFFFFFFFFFF0F0F0F0AA55
AA55AA55AA5500000000AA55AA55AA55
AA55000000000024240000000001010
20000000000000083E283E0A3E080000
001000001000003C4204080008000004
08080808040000201010101020000000
10080408100000000408100804000000
003E003E0000000008083E0808000000
00003E00000000014083E08140000000
02040810200000001000001010200000
000000080810000000000181800003C
464A52623C000018280808083E00003C
42023C407E00003C420C02423C000008
1828487E0800007E407C02423C00003C
407C42423C00007E020408101000003C
423C42423C00003C42423E023C000018
24427E424200007C427C42427C00003C
```

CHECKSUM = 15684

START = 17474

```
424040423C000078444242447800007E
407C40407E00007E407C40404000003C
42404E423C000042427E424242000008
0808080808000002020242423C000044
4850684442000040404040407E000082
C6AA92828200004262524A464200003C
424242423C00007C42427C404000003C
4242524A3C02007C42427C444200003C
403C02423C0000FE1010101010000042
424242423C0000828282442810000082
828292AA440000422418182442000082
442810101000007E040810207E000000
000000000000F0F0F0F000000000F0F
0F0F00000000FFFFFFF0000000000000
0000F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F
0F0FF0F0F0F0FFFFFFFFFFF0F0F0F0AA55
AA55AA55AA5500000000AA55AA55AA55
AA55000000000024240000000001010
200000000000083E283E0A3E080000
001000001000003C4204080008000004
```

CHECKSUM = 20782

Also shown are big asterisks (*) indicating which option was selected. The actual start of the program is signalled by a rather garish screen display, which is also used at the end to signal completion. A somewhat less flashy display signals the completion of each problem.

In the options you've installed so far, a large letter or number is flashed on the screen, and your pupil must find the corresponding key on the keyboard, and press it.

The philosophy here is not to tally "right" vs. "wrong" responses, so the program will persist until it gets the right answer. If the wrong key is pushed, a "sad face" is displayed and the program flashes, "NOT ..., TRY AGAIN." Coaching will be required of you, reading the messages aloud until the child recognizes them himself. Obviously, a child who is learning the alphabet can't be expected to know how to read; verbal prompts are therefore kept to a bare minimum. It will also take outside work to learn the phonetic names of the characters. This program is not a baby-sitter, at least not at the outset. It is intended to supplement parental guidance, not to replace it.

Also shown, in the top left corner, is which problem the student is working on (1-26 for the alphabet drills, or 1-10 for the numbers drill of option 3). This is both for your information as you monitor the child's progress, and also to re-inforce the significance of counting.

Pressing the right key merits (you guessed it) a "happy face." The problem then scrolls out of view, and the next character is displayed.

Scoring is done as follows: while the program is waiting for a response, it continually "counts time." At the end of the set of problems, the average response time (time required to get the right answer) is computed. Depending on the response time, from zero to 10 "robots" are displayed. A perfect score (10) represents an average response time of two seconds. Each unit (robot) less,

```

1 GOTO 8000
2 FOR I=1 TO LEN M$
3 PRINT AT 1,0,I
4 POKE CH,CODE M$(I)
5 IF USA BT THEN REM <>>
6 LET C=C+N4
7 IF C<32 THEN NEXT I
8 RETURN
9 REM *****
10 REM *****
11 REM *****
12 REM *****
13 REM *****
14 REM *****
15 REM *****
16 REM *****
17 REM *****
18 REM *****
19 REM *****
20 LET A$=INKEY$
21 LET D=D+N1
22 IF A$<"0" OR A$>"L" THEN GOT
23 20
24 RETURN
25 REM *****
26 REM *****
27 FOR I=0 TO LEN
28 NEXT I
29 RETURN
30 REM *****
31 REM *****
32 POKE 16418,0
33 PRINT AT 23,0,"
34
35 POKE 16418,N2
36 FOR I=1 TO 24
37 IF USA SCR THEN
38 PRINT AT 21,31,"
39 NEXT I
40 RETURN
41 REM *****
42 FOR S=1 TO 30
43 LET Y=(S/5)*3+4
44 LET Q=3+S
45 IF Q>25 THEN LET Q=Q-25
46 PRINT AT Y,Q,"<"TAB Q;"
47 "TAB Q;"<"TAB Q;"<"TAB
48 "TAB Q;"<"TAB Q;"<"TAB
49 "TAB Q;"<"TAB Q;"<"TAB
50 NEXT S
51 RETURN
52 REM *****
53 REM *****
54 REM *****
55 REM *****
56 REM *****
57 REM *****
58 LET SC=INT ((11-D)/(10*LEN T$
59 ))
60 IF SC=0 OR SC=-N1 THEN LET
61 SC=N1
62 IF SC<-N1 THEN LET SC=0
63 GOSUB 200
64 PRINT AT 21,0:
65 GOSUB (390+20*INT (.7+SC/3)
66 )
67 RETURN
68 PRINT "NO GOOD, ";N$;". TOO
69 SLOW."
70 RETURN
71 PRINT "TRY AGAIN, ";N$
72 RETURN
73 PRINT "PRETTY GOOD ";N$;".
74 KEEP GOING"
75 RETURN

```

START = 17794

```

08080808040000201010101020000000
10080408100000000408100804000000
003E003E0000000008083E0808000000
00003E000000000014083E0814000000
02040810200000001000001010200000
0000000808100000000000181800003C
464A52623C000018280808083E00003C
42023C407E00003C420C02423C000008
1828487E0800007E407C02423C00003C
407C42423C00007E020408101000003C
423C42423C00003C42423E023C000000
38043C443C0000404078444478000000
384440403800004043C44443C000000
38447C403800001C2078202020000000
3844443C043800404078444444000010
0030101038000080018080848300020
24283028240000301010101038000000
68545454440000058644444444000000
3844444438000007844447840400000
3C44443C040600002C32202020000000

```

CHECKSUM = 9684

START = 18114

```

3C403C023C0000207820202418000000
444444443C0000004444442810000000
44545454280000004428102844000000
41221408083000007C0810207C000000
00003A3B40F5CD2B0FCD2549A72002CF
FFFE2820F4F1CB77C0C3230F444DC335
0F434DC3F5087D010000C3750BC50610
7C4D21000029CB11730011910F7C1C9
0E0118060E0218020E007CAAE68047D5
CD9D47E3CD9D47EBE179A7283678B1E5
ED62ED52E5C1D1210000373FF57A531E
082930058FED4A180987ED4A3803ED42
3D3C1D20EC5FF130E2CB4F2001EBCB7F
C8180BCD2F47CB1D0D18037C07D0EBA7
ED62ED52C97C073806B5C8210100C921
FFFFC9A7ED52C9A7ED527CB5210000C0
2CC9A7ED52C818E3EB0100801804EB01
01807CA8677AA857ED522600796FD8EE
016FC97CB5210000C87AB3C82CC97CB5
B2ED6218F53E011801AFE35E235623E3

```

CHECKSUM = 30335

START = 18434

```

2B2919A72808EBE1E3EB732372C95E23
56EBC9EBCD1B48EBC9D5114B00ED4B32
40036268CD0913E56269CD0913C17C81
300104677D903001256F2B223240D1C9
3A2140CB4720197C0730093E16D7EBA7
ED62ED5201F0D81EFC5CDE107CDAD0A
E5C1215D40571E1C70CB7A28017197ED
6F83D797ED6F83D7CB7ACBFA28EF72C9
E17E23FE70282FFE762003D718F3CD29
4718EE9718023E904B452130405677D5
E5CDB20BELD172C9CDBC477CB5E35E23
5623E3C0E1EBE97DE61F4FFDCB014E28
0AFD9638CBFFC63CD47108FD8639FE21
3A3A40DE01CDFA08FDCB01C6C9E15E23
5623E5D5210700190604562B5E2BD510
F9E1C1D17A19EBE1E3732372C5E10730
01EBCDCB47E1D8C1E9EB73C9C1E5C1C9
6E2600C911DA7FD5E5C9C5E1C9EBCD2F
49EBC9CD2F4938FBCD2F4930FBD5CDBB
02E5C12C6C2804CDBD076E26007DD1C9

```

CHECKSUM = 40606

```

450 PRINT "VERY,VERY GOOD, ";N$
460 RETURN
470 PRINT AT 20,0;"*10* YAY, "
;N$;" *TOP SCORE*"
480 RETURN
500 REM *****
510 PRINT AT 16,0;
520 PRINT "
530 PRINT "
540 PRINT "
550 PRINT "
560 PRINT "
570 PRINT "
580 RETURN
600 REM *****
610 PRINT AT 16,0;
620 PRINT "
630 PRINT "
640 PRINT "
650 PRINT "
660 PRINT "
670 PRINT "
680 RETURN
700 REM *****
710 PRINT AT 16,0;
720 FOR I=N1 TO 6
730 PRINT "
740 NEXT I
750 RETURN
800 REM *****
810 POKE FILL, CODE "
820 IF USR UP THEN
830 POKE FILL, CODE "
840 IF USR UP THEN
850 POKE FILL, CODE "
860 IF USR UP THEN
870 POKE FILL, CODE "
880 IF USR UP THEN
890 POKE FILL, 0
900 IF USR UP THEN
910 RETURN
920 REM *****
930 POKE LN, 24
940 IF USR REV THEN
950 FOR I=N1 TO 10
960 IF INKEY$<>"" THEN RETURN
970 NEXT I
980 GOTO 940
1000 REM *****
1010 LET D=0
1020 FOR N=N1 TO LEN T$
1030 LET C$=T$(N)
1040 PRINT AT 0,0;N
1050 LET R=10
1060 LET C=14
1070 LET M$=C$
1080 GOSUB N2
1090 GOSUB 20
1100 IF A$=C$ THEN GOTO 1220
1110 GOSUB 600
1120 LET R=10
1130 LET C=14
1140 LET M$=""
1150 GOSUB N2
1160 PRINT AT 11,13;"NOT ";A$;AT
1170 LET LEN=30
1180 GOSUB PA
1190 GOSUB 700
1200 PRINT AT 11,13;" ";AT 1
1210 GOTO 1040
1220 GOSUB 500
1230 LET LEN=25
1240 GOSUB PA
1250 GOSUB 150
1260 GOSUB 850
1270 NEXT N
1275 GOSUB 300
1280 GOSUB 300
1290 GOSUB 920
1300 GOSUB 150
1310 POKE 16752,67
1320 IF SC>7 THEN GOTO 1380
1330 PRINT AT N1,N2;
1340 IF USR BT THEN REM <>0-9
1350 PRINT AT N4,N2;
1360 IF USR BT THEN REM <>0-9
1370 GOTO 9640
1380 IF (F$="2" AND Y$="L") OR F$="3" THEN GOTO 8800
1390 IF F$<>"2" THEN GOTO 1430
1400 LET F$="1"
1410 LET Y$="L"
1420 GOTO 1440
1430 LET F$="2"
1440 GOSUB 3000
1450 GOTO 9600
2000 REM *****
2010 CLS
2020 PRINT "NOT INSTALLED YET"
2030 STOP

```

represents two additional seconds average. (You can modify this, as indicated in the REM in the program listing.)

You may wish to call the little creatures "robots" or "monsters." If the child watches CTV's "Sesame Street," he might prefer the term "monster" because of the positive connotation of the word (thanks to Cookie, Grover and the gang).

Options 1 and 2 can be used to drill either upper-case (capital) or lower-case letters. You will be prompted appropriately from the "Parent/Teacher Section." Normally, you would start with capitals.

These options follow a natural progression of levels. When the child has mastered one level (by getting at least 7 robots), the program sequences to the next level, in the following order:

Alphabet Trainer (capitals)
 Alphabet Scramble (capitals)
 Alphabet Trainer (lower-case)
 Alphabet Scramble (lower-case)

When the entire sequence is completed, the program flashes "ALL DONE" and stops. Enter RUN to start afresh, or GOTO 9250 to keep the same name.

1. Alphabet Trainer

This sequentially flashes the letter A-Z. Its purpose is to relate the letters on the screen with those on the keyboard, and to teach the location of the keys. Young children will use the "one-finger" approach, but there's no reason why the same program couldn't be used to help older children learn to "touch-type" with eight fingers.

The capital letters are, for the most part, blow-ups of the character patterns in the ROM. The exceptions are letters (such as A and I) which look different on-screen than they do on the key legends. The patterns for Q, W and V were also modified to make them more easily recognized. The lower-case patterns are the more-or-less "typical" computer typeface.

2. Alphabet Scramble

This is just like the trainer, except that the letters are scrambled up, so that there will be a different sequence every time. Letters are never repeated in a given set, however, and all 26 letters appear each time through. When selecting (or progressing to) a "scramble" option, the screen will briefly go blank while the program shuffles the characters.

3. Numbers Scramble

This option is very similar to #2, except that it drills the ten numeric digits instead of the alphabet. Since the number keys are in sequence on the keyboard, there is not much point in having a "trainer" mode. Scoring is identical to the alphabetic mode.

Coaching Hints

This collection of programs follows the philosophy of the old saw, "Try, try again." Re-inforce this with supportive coaching; if the student becomes discouraged by repeated sad faces, build up his enthusiasm with encouraging remarks. Have him "find the button that looks like that (letter) on the TV."

When he gets it right, give him an enthusiastic commendation or (better yet) a hug or other physical reward.

At the end, read him the message that appears with the robot "score." These range from "TOO SLOW" to "YAY, YOU GOT THE TOP SCORE."

These drills (especially the alphabet) can tax a child's attention span. Take a break for cookies and milk, or another activity, between "sets." The computer can wait, patiently flashing the screen until your pupil returns for another go.

Make sure that the child understands that high scores result from fast response. If he starts to dawdle, a comment like "Hurry up, so you'll get lots of monsters" might help. If you have to do it too often, though, take it as a sign that the child is losing interest. Help him finish that set, then relieve the monotony with another activity.

The BREAK key will stop operation. He'll let you know, perhaps tearfully at first, that "the computer broke." Assure him that all is well, and get it going again by deleting whatever garbage he managed to get on the input line, then enter CONT. After a while, the pupil will catch on and be able to do this himself.

```

3000 REM *****
3010 LET T$="0123456789"
3020 IF F$<"3" THEN LET T$="ABCD
      EFGHIJKLMNOPQRSTUVWXYZ"
3030 LET L$=T$(LEN T$)
3040 IF F$="1" THEN RETURN
3050 FAST
3060 GOSUB 3300
3070 IF F$<"4" THEN GOTO 3130
3080 IF F$>"4" THEN GOTO 3150
3090 LET S$=""
3100 FOR I=N1 TO 10
3110 LET S$=S$+CHR$ (ST+28)
3120 NEXT I
3130 SLOW
3140 RETURN
3150 LET S$=T$
3160 GOSUB 3300
3170 SLOW
3180 RETURN
3300 REM *****
3310 LET C=LEN T$
3320 FOR I=1 TO N4+C
3330 LET R=N1+INT (RND+C)
3340 LET T$=T$(R)+T$( TO R-N1)+T
      $(R+N1 TO )
3350 NEXT I
3360 RETURN
4000 REM *****
4010 GOTO 2000
8000 REM *****
8010 POKE LN,24
8020 IF USR REV THEN
8030 PRINT AT N3,N2;
8040 IF USR BT THEN REM <>0 ALL
8050 PRINT AT 9,N4;
8060 IF USR BT THEN REM <>0 DONE
8070 PRINT AT 16,0;
8080 IF USR BT THEN REM <>0
8090 LET C=16-2*LEN N$
8100 LET R=10
8110 LET M$=N$
8120 GOSUB N2
8130 GOSUB 920
8140 GOSUB 800
8150 PRINT AT 9,N2;
8160 IF USR BT THEN REM <>0THE E
      ND
8170 PRINT AT 0,0;
8180 GOSUB 520
8190 STOP
8195 GOTO 9250
9000 REM *****
9010 LET X$=" "
9020 POKE 16752,67
9030 LET BT=16601
9040 LET UP=16928
9050 LET FILL=16929
9060 LET REV=17040
9070 LET LN=17044
9080 LET SCR=17060

```

```

9090 LET CH=21609
9100 LET PA=100
9110 LET O=0
9120 LET N1=1
9130 LET N2=2
9140 LET N3=3
9150 LET N4=4
9160 LET N5=5
9165 LET HIGH=0
9170 SLOW
9180 PRINT AT 0,0;
9190 IF USR BT THEN REM <>0
9200 PRINT AT N1,N1;"THE";AT N4,
      N2;
9210 IF USR BT THEN REM <>0PROGR
      AM
9220 PRINT AT 8,5;"*****
      *****"
9230 INPUT N$
9240 IF LEN N$>8 OR LEN N$<N1 TH
      EN GOTO 9230
9250 PRINT AT 10,12;N$;TAB 12;"
      " ( TO LEN N$)
9260 PRINT "SELECT AN OPTION:"
      TAB N5;"1. ALPHABET TRAINER"
      TAB N5;"2. ALPHABET SCRAMBLE"
      TAB N5;"3. NUMBERS SCRAMBLE"
      TAB N5;"4. ADDITION DRILL"
      TAB N5;"5. ADDITION SCRAMBLE I"
      TAB N5;"6. ADDITION SCRAMBLE II"
      TAB N5;"7. ALPHA-SPACE BRIDGE"
9270 LET F$=INKEY$
9280 IF F$<"1" OR F$>"7" THEN GO
      TO 9270
9290 PRINT AT 14+VAL F$,N4;" "
9300 IF F$="7" THEN GOTO 9460
9310 IF F$<"4" THEN GOTO 9450
9320 CLS
9330 PRINT "DRILL WHICH NUMBER (
      0-9)?"
9340 INPUT ST
9350 IF ST<0 OR ST>9 THEN GOTO 9
      340
9360 LET ST=INT ST
9370 IF ST<9 THEN GOTO 9400
9380 LET FI=ST
9390 GOTO 9450
9400 PRINT ST,"LAST NUMBER TO
      DRILL ("ST,"-9)?"
9410 INPUT FI
9420 IF FI<ST OR FI>9 THEN GOTO
      9410
9430 LET FI=INT FI
9440 PRINT FI
9450 GOSUB 3000
9460 POKE 16418,0
9470 PRINT AT 22,0;"*****
      *****"
9480 IF INKEY$<" " THEN GOTO 948
      0
9490 IF INKEY$="" THEN GOTO 9490
9500 LET Y$="U"
9510 IF F$>"2" THEN GOTO 9570
9520 PRINT AT 22,0;"UPPER (CAPIT
      ALS) OR LOWER CASE?"
9530 IF INKEY$<" " THEN GOTO 953
      0
9540 IF INKEY$="" THEN GOTO 9540
9550 LET Y$=INKEY$
9560 IF Y$<"U" AND Y$<"L" THEN
      GOTO 9550
9570 PRINT AT 22,0;"READY FOR P
      UPIL "+N$+" " ( TO
      32)
9580 POKE 16418,N2
9590 GOSUB 920
9600 CLS
9610 POKE 16752,67
9620 PRINT AT N3,N4;
9630 IF USR BT THEN REM <>0
9640 LET M$=N$
9650 LET R=10
9660 LET C=16-2*LEN M$
9670 GOSUB N2
9680 POKE 16752,(67 AND Y$="U")+
      (69 AND Y$="L")
9690 LET M$="*****" ( TO VAL F$
      )
9700 LET R=16
9710 LET C=16-2*LEN M$
9720 GOSUB N2
9730 GOSUB 920
9740 GOSUB 800
9750 RAND
9760 IF F$="7" THEN GOTO 4000
9770 IF F$<"4" THEN GOTO 1000
9780 GOTO 2000
9990 SAVE "ABC12"
9994 RUN
9995 REM
9996 REM
9997 REM
9998 REM
9999 REM

```

HANG—MULT 2068

by Kent E. Cook

The mainframe where I work has a Hangman-style number guessing game in its library which has fascinated me. I wrote HANG-MULT to emulate the mainframe's program on my TS-2068 and thought other SWN readers might enjoy it too.

The program's title screen includes a PAUSE to prevent the random-number SEED from starting at the same value each time the program is loaded.

After the title screen prompt is answered, the program creates a random 3-digit and 2-digit number which are multiplied and converted to a subscripted string variable. Another string is created and used to display the problem. Unidentified digits are represented as "X". Previous guesses are listed by column. The program then displays prompts for column, number, and yes or no.

The data-input request prompts may be answered with digits 1 through 5 for "Column =" (5 is the most significant column), 0 through 9 for "Number =", and "Y", "y", or ENTER for an affirmative response to "OK? (Y/N)" (anything else for a negative). These prompts may also be answered with "?" to forget. The solution will then be displayed and all variables CLEARED. All prompts, except the title page, may be answered with "S" or "s" to SAVE the program to tape. After the SAVE, program execution continues at the point the SAVE occurred. When reloaded, the program retains the problem data existing at the time of the SAVE (I.E. None if SAVED from the forget screen).

Guessing all 18 numbers displays the solved problem, a rating, and statistics. A prompt offers "Another game (Y/N)?". Exiting from this prompt CLEARS the variables. All variables and the UDG are defined in the program, so SAVE "HANG-MULT" LINE 0 is OK, as is RUNNING to re-start.

There is an error trap which sends the program to the exit routine and CLEARS everything ON ERROR. This prevents the temptation of BREAKING into the program and changing data.

The only thing in the listing that's unclear is the GRAPHICS character defined in line 1090 and printed in line 210. It doesn't matter which character is used providing it's the same in both places. I used GRAPHICS "I" (for "line").

```

10 REM COLD START
20 ON ERR GO TO 1900: GO SUB 3
00: LET gm=0: LET tm=0: PRINT "
Press any key to begin": PAUSE
0: GO TO 1000
30 DEF FN r$(x)=STR$(INT((100
%x)/100))+
100 REM DATA INPUT
110 BORDER 0: ON ERR GO TO 120
120 IF INKEY$<>"" THEN STOP
130 PAUSE 0: LET I$=INKEY$: IF
I$="S" OR I$="s" THEN SAVE "HANG
-MULT" LINE 1
140 GO SUB SUB5: ON ERR GO TO 19
00: RETURN
200 REM PRINT PROBLEM
210 PRINT AT 0,10:"HANG-MULT"
FOR x=1 TO 5: PRINT TAB 12,d
$(x): IF x=2 OR x=4 THEN PRINT T
AB 12,"LLLLL"
220 NEXT x: RETURN
240 NEXT y: NEXT x: PRINT AT 19
0:"Misses: "im:" ": RETURN

```

```

300 REM *****
410 GO SUB 400: PRINT AT 6,0;"
*****" 1986""Kent E.
Cook""PO Box 1031""Terre Haute
Ind.""47808": RETURN
400 REM *****
410 INK 7: PAPER 0: BORDER 0: C
LS: RETURN
500 REM *****
510 PRINT #1,AT 0,0;"

```

```

1000 REM *****": RETURN
1010 LET gm=gm+1: LET hit=0: LET
mt=0: DIM c$(5,20): DIM a$(5,5)
: LET a$(1)="0000": LET a$(2)="
0000": LET a$(3)="00000": LET
$(4)="00000": LET a$(5)="00000"
: DIM d$(5,5): LET d$(1)="XXX":
LET d$(2)="XX": LET d$(3)="
XXXX": LET d$(4)="XXXX": LET d$(
5)="XXXXXX"
1020 RANDOMIZE 0: LET x=INT (1+R
ND*9999): LET a$(1,6-LEN STR$ x T
O)=STR$ x
1030 LET x=INT (1+RND*999): LET a
$(2,6-LEN STR$ x TO)=STR$ x
1040 LET x=VAL a$(1)+VAL a$(2,5)
: LET a$(3,6-LEN STR$ x TO)=STR
$ x
1050 LET x=VAL a$(1)+VAL a$(2,4)
: LET a$(4,5-LEN STR$ x TO)=STR
$ x: LET a$(4,5)=" "
1060 LET x=VAL a$(1)+VAL a$(2):
LET a$(5,6-LEN STR$ x TO)=STR$
x
1070 DIM p$(160): FOR x=0 TO 4:
LET p$(32*x+1 TO)="Col "+STR$ (
5-x)+": NEXT x
1080 REM *****
1090 GO SUB 400: FOR x=0 TO 7: P
OKE USR "L"+x,(255 AND x=4): NEX
T x: LET L=1080

```

AT LAST! A VERY AFFORDABLE COMPUTER AT A VERY AFFORDABLE PRICE

POWERFUL FULLY PROGRAMMABLE WITH 2K OF MEMORY—PORTABLE—6-7/8" x 1-3/8" INCH MODULE SINGLE-KEY ENTRY COMMANDS—DURABLE 40 KEY MEMBRANE TYPE KEYBOARD—280A BASED FOUR CHIP DESIGN—EDUCATIONAL—UNIQUE SYNTAX-CHECK REPORT CODES FOR ERROR IDENTITY—GRAPH DRAWING AND ANIMATED DISPLAY—ACCURATE TO 9/12 DECIMAL PLACES FOR FULL RANGE MATH AND SCIENTIFIC FUNCTIONS—AT AN AFFORDABLE PRICE

WE CANNOT TELL YOU THE MAKE OF THE COMPUTER BUT IT WAS MADE BY A FAMOUS WATCH COMPANY THAT USED TO SELL FOR \$99.95

WE BOUGHT OUT WHAT THE FACTORY HAD LEFT IN STOCK AND HAD TO REMOVE THE LABELS. THESE UNITS ARE UNPACKAGED LESS THE IV WALL ADAPTER AND MANUAL BECAUSE THIS IS A DISCONTINUED ITEM THERE IS NO WARRANTY

GET THEM WHILE THEY LAST

LIMITED SUPPLY

BUY 1st UNIT FOR \$19.95 BUY 2nd FOR \$19.95 9V DC WALL ADAPTOR \$4.95
BUY THE 3rd UNIT (non operating for parts) \$10.95 MANUAL (over 150 pages) \$2.95

See September 1984 issue of 73 for TIME/RTTY article

CHIP BONANZA (AT THESE PRICES THEY ARE A STEAL)

| | | |
|--|-----------------|-----------------------------|
| 2708 | | \$1.00 EA OR 10 FOR \$ 9.00 |
| 2718 | | \$2.25 EA OR 10 FOR \$20.00 |
| 2732 | | \$3.25 EA OR 10 FOR \$30.00 |
| 2784 | | \$4.00 EA OR 10 FOR \$35.00 |
| 27138 | | \$7.00 EA OR 10 FOR \$65.00 |
| 6502 | | \$4.95 EA OR 10 FOR \$45.00 |
| 6810 | (REG. \$3.00) | \$1.35 EA OR 10 FOR \$18.00 |
| 68A08 | (REG. \$10.95) | \$5.95 EA OR 10 FOR \$50.00 |
| 68A21 | (REG. \$9.95) | \$2.95 EA OR 10 FOR \$25.00 |
| 4134 | (or equivalent) | 8 FOR \$ 5.99 |
| | | 9 FOR \$12.95 |
| XR2121 | (SPECIAL) | \$ 2.95 |
| 25S 9200XL MICRO-P 64 PIN 8 BIT D/B | > 16 BIT CPU | \$ 4.95 |
| 25S 9201XL MICRO-P PS1 | | \$ 2.95 |
| TM1 9018AL MICRO-P CLOCK GEN AND DRIVER | | \$ 5.95 |
| TM5 9018AL MICRO-P COLOR GRAPHICS AND DISPLAY | | \$ 9.95 |
| KEYBOARD (6804) 48 KEYS MEASURE 4 x 10 (H-TEQ) | | \$ 9.95 |

APPLE II & APPLE II+ COMPUTER
MAINFRAMES (fully populated) \$150.

Power supply, case and keyboard, separately available Call or Write

As up as described above, fully assembled & tested \$295.00 plus shipping

APPLE POWER

SUPPLIES \$29.95

Cassette Software:

I have cassette software
..... send for list (reg. \$9.95-\$14.95)
Hal's Special Price: 1 for \$3.95;
3 for \$10.95; 6 for \$19.50
10 for \$29.50; 20 for \$50.00
..... or let Hal select 25 different cas-
settes (your choice) at \$2.95/00.00!!!

SHIPPING INFORMATION: PLEASE INCLUDE 10% OF ORDER FOR SHIP-
PING AND HANDLING CHARGES (MINIMUM \$2.50, MAXIMUM \$10). CA-
NADIAN ORDERS, ADD \$7.50 IN US FUNDS. MICHIGAN RESIDENTS ADD
4% SALES TAX. FOR FREE FLYER, SEND 22¢ STAMP OR SASE.

HAL-TRONIX, INC.

P.O. BOX 1101 DEPT. N
12671 DIX-TOLEDO HWY
SOUTHGATE, MICH. 48195

~ 19

PHONE (313) 281-7773

SPECIALS

IBM Compatible Computers

Flip-Top Case, Keyboard,

Power Supply, and Mother-

board. Up and Running

\$399.00 complete (plus

\$5.00 for shipping)

FCC Approved

CALL FOR DETAILS

LNA

90° Kevlin, self-contained

electronic polarator, and

feedhorn \$59.95 each

or 2 for \$100.00 (plus \$5.00

for shipping)

"HAL"

HAROLD C. NOWLAND
W8ZXH


```

1100 REM DISPLA PROBLEM
1110 GO SUB 200: IF hit=18 THEN
GO TO 1500
1120 PRINT AT 10,0:"Previous gue
sses: ";P$;"Game #";gm;"Misses
=";mt;"Average Misses/game=";
FN r$(mt+tm)/gm)
1200 REM INPUT GUESS
1210 PRINT #1;AT 0,0:"Column = ?
": GO SUB 100: IF I$="" THEN GO
TO 1400
1215 IF I$<"1" OR I$>"5" THEN GO
TO 1210
1220 LET col=VAL I$
1230 PRINT #1;AT 0,0:"Column = "
;col;"Number = ?": GO SUB 100: I
F I$="" THEN GO TO 1400
1235 IF I$<"0" OR I$>"9" THEN GO
TO 1230
1240 LET n=VAL I$
1250 PRINT #1;AT 0,0:"Column = "
;col;"Number = ";n;"OK (Y/N)?":
GO SUB 100: GO SUB 500: IF I$=""
THEN GO TO 1400
1255 IF CODE I$>13 AND I$<"Y"
AND I$<"y" THEN GO TO 1200
1260 LET col=6-col: FOR x=2 TO 2
0 STEP 2: IF c$(col,x)<>STR$ n T
HEN NEXT x: GO TO 1300
1265 PRINT #1;AT 0,0:" You alrea
dy tried that!
": BEEP .5
": PAUSE 40: GO SUB 500: GO TO
1200
1300 REM MISS
1310 LET miss=1: FOR x=20 TO 2 5
TEP -2: IF c$(col,x)=" " THEN NE
XT x: LET c$(col,2)=STR$ n: GO T
O 1340
1320 FOR x=x TO 2 STEP -2: IF c$
(col,x)>STR$ n THEN NEXT x

```

```

1330 LET c$(col,x+2 TO )=STR$ n+
"+c$(col,x+2 TO 18)
1340 FOR x=0 TO 4: LET y=32*x+8:
LET p$(y TO y+20)=c$(x+1): IF a
$(x+1,col)=STR$ n THEN BEEP .1,3
6: LET hit=hit+1: LET d$(x+1,col
)=STR$ n: LET miss=0
1350 NEXT x: LET mt=mt+miss: GO
TO 1100
1400 REM FORFEIT
1410 GO SUB 400: FOR x=1 TO 5: L
ET d$(x)=a$(x): NEXT x: GO SUB 2
00: PRINT "Previous game reco
rds erased!""Start over (Y/N)?
": LET l=0: GO SUB 100: IF I$<>
"N" AND I$<>"n" THEN RUN
1420 GO TO 1900
1500 REM SOLVE IT
1510 GO SUB 400: GO SUB 200: LET
tm=tm+mt: PRINT "You solved i
t with ";mt;" misses";"es" AND mt<
>1:"-"-"An unbeatable performan
ce" AND mt=0:"A world class perf
ormance!" AND (mt=1 OR mt=2):"A
first class performance" AND (mt
=3 OR mt=4):"A good effort!" AND
mt=5:"Not bad for a tough probl
em" AND (mt=6 OR mt=7):"Too bad!
This one was tricky" AND (mt=8
OR mt=9):"A poor showing, but ke
ep trying" AND mt>=10 AND mt<=12
:"I hope no one is watching!" AN
D mt>=13
1520 PRINT "Average misses/gam
e after ";gm;" game";"s" AND gm<
>1:"=";"FN r$(tm/gm)/"Another
game (Y/N)?": GO SUB 100: IF I$<
>"N" AND I$<>"n" THEN GO TO 1000
1900 REM END
1910 CLEAR : GO SUB 300: ON ERR
RESET: STOP

```

BASIL'S COMPENDIUM

Basil Wentworth

GO TO / GO SUB

This chapter will introduce the concepts of JUMP and CALL, the machine code analogs of BASIC's GO TO and GO SUB.

Practitioners of machine code have their own names for everything, it seems. When it comes to branching, they use the term JUMP, rather than GO TO, and CALL instead of GO SUB. Maybe it sounds more dynamic that way.

Shakespeare used the term "Go to!" by itself. (From which speakers of Hebrew will understand why MacBeth said "Lay on, MacDuff!") In later days, everyone, including computer programmers, has found it advisable to follow the expression with the name of a destination--albeit sometimes one that is physically unreachable. A GO TO or GO SUB expression in BASIC has to be followed by a line number. (Or by an expression that the computer can interpret as a line number. For instance, GO TO A is a perfectly acceptable instruction, if A has been defined as a number that the computer could reasonably interpret as a line number.) In the case of machine code, where there are no line numbers, the JUMP address is given in terms of bytes. The address can be expressed either as

1. The byte number (memory location) to which the program is to JUMP, or
2. The number of bytes (either forward or back) that the program is to JUMP.

In either case, you will want to have the program laid out neatly, so that you can count the bytes. I find it easiest to write my rough drafts of a program on ruled paper, one byte per line, as shown in Figure 11-1. If you want to save paper by arranging the chart in two (or more) columns, go ahead, as long as the paper is wide enough for you to read all the columns without confusion. You'll recognize this program as our old friend 2+2, expressed in a new way.

| | | | |
|-------|-----|------|--------|
| 16514 | 62 | Y | LD A,2 |
| 16515 | 2 | Gr 2 | |
| 16516 | 60 | W | INC A |
| 16517 | 60 | W | INC A |
| 16518 | 6 | Gr T | LD B,0 |
| 18519 | 0 | | |
| 16520 | 79 | ? | LD C,A |
| 16521 | 201 | TAN | RET |

Figure 11-1. 2+2 in Machine Code

Once you get the code into P\$ at line 100, and RUN the program, the proofreading stage will give you just such a listing, although without the mnemonics. Then you can insert a JUMP command wherever you want it, followed by the JUMP destination in 256-bit (taking into account the presence of the JUMP instruction bytes, of course).

For instance, if you want to skip one of the INC A stages of Figure 11-1, your program could look like Figure 11-2.

I repeat--note that the calculation of your JUMP destination must take into account the three bytes of the JUMP instruction itself--unless, of course, you're JUMPing backwards.

A word of caution: when you do choose to JUMP backwards, be careful that you don't fall into an endless loop. If Fig. 11-2 had said JP 16516 or JP 16517, for instance, the program would have crashed. Do you see why? Follow it through, byte by byte, for a couple of cycles, and you'll get the idea.

| | | |
|-------|---------|----------|
| 16514 | 62 Y | LD A,2 |
| 16515 | 2 Gr 2 | |
| 16516 | 60 W | INC A |
| 16517 | 195 ? | JP 16521 |
| 16518 | 137 ? | |
| 16519 | 64 RND | |
| 16520 | 60 W | INC A |
| 16521 | 6 Gr T | LD B,0 |
| 16522 | 0 | |
| 16523 | 79 ? | LD C,A |
| 16524 | 201 TAN | RET |

Figure 11-2. Skipping one of the INCs

The example in Fig. 11-2 is absurd, of course. It would have been far easier to delete the extra INC byte than to JUMP over it. But my purpose at this stage is to show you how to do things when you want to. I offer no guarantee that I'll give you a good reason to want to do them.

A slightly more rational program can be used to demonstrate the CALL instruction. Like its BASIC analog, GO SUB, this instruction is most often used when you have to repeat an operation several times. Suppose your program had called for 2+2+2+2+2. You could have inserted the INC instruction 8 times. However, you could also do it by putting the +2 operation into a subroutine, as in Fig. 11-3.

| | | |
|-------|-----------|-----------------------|
| 16514 | 62 Y | LD A,2 |
| 16515 | 2 Gr 2 | |
| 16516 | 205 LN | CALL 16532 |
| 16517 | 148 Inv = | |
| 16518 | 64 RND | |
| 16519 | 205 LN | " |
| 16520 | 148 Inv = | |
| 16521 | 64 RND | |
| 16522 | 205 LN | " |
| 16523 | 148 Inv = | |
| 16524 | 64 RND | |
| 16525 | 205 LN | " |
| 16526 | 148 Inv = | |
| 16527 | 64 RND | |
| 16528 | 6 GrT | LD B,0 |
| 16529 | 0 | |
| 16530 | 79 ? | LD C,A |
| 16531 | 201 TAN | RET (to BASIC) |
| 16532 | 60 W | INC A |
| 16533 | 60 W | INC A |
| 16534 | 201 TAN | RET (from subroutine) |

Figure 11-3. The Subroutine

Notice that the RETURN from a subroutine (201 in byte 16534) is the same instruction as the RETURN to BASIC (byte 16531). Notice also that you need both of them. Unless you manage to make one of them do double duty, as in Fig. 11-4.

The JUMP RELATIVE instruction (JR) works very much like the JUMP, except that it tells the program how far to JUMP, rather than giving it a destination. Think of it as a local resident giving instructions to a visiting motorist. He could say "Turn right on Main Street" (JUMP), or he could tell you to "Go

| | | |
|-------|-----------|------------|
| 16514 | 1 Gr 1 | LD BC,2 |
| 16515 | 2 Gr 2 | |
| 16516 | 0 | |
| 16517 | 205 LN | CALL 16526 |
| 16518 | 142 Inv : | |
| 16519 | 64 RND | |
| 16520 | 205 LN | " |
| 16521 | 142 Inv : | |
| 16522 | 64 RND | |
| 16523 | 205 LN | " |
| 16524 | 142 Inv : | |
| 16525 | 64 RND | |
| 16526 | 3 Gr 7 | INC BC |
| 16527 | 3 Gr 7 | " |
| 16528 | 201 TAN | RET |

Figure 11-4. Doubling up the RETURN

East for five blocks, and then turn right" (JUMP RELATIVE). The JR instruction takes up only two bytes instead of the three required for JP, and it has the advantage that subsequent changes in the program are not so likely to mess things up as they are with JUMP. Moreover, you can more easily relocate the whole routine if you want to, if you've used JR's.

But don't get overconfident. Any time you make ANY change in the program, check it over to make sure that JUMP destinations and JR distances are still good. Just as you would check all GOTO and GOSUB addresses if you change any line numbers in BASIC.

The JR instruction is 24, followed by one byte giving the DISPLACEMENT, or the distance to jump. The jump can be either forward (positive) or back (negative). For negative numbers, we use the convention mentioned earlier: 255=-1, 254=-2, etc. Obviously, the positive and negative numbers are going to meet at 128, so don't plan to jump more than 127 bytes in a single hop. If you ever do have to make a JR more than 127 bytes long, you can do it in 2 or more hops, jumping first 100 bytes, for example, landing you on an instruction to JR an additional 50. (Sounds a little like a game of Monopoly.)

One instruction to avoid at all costs is JR 254 (=JR-2). Do you see why? (HINT: remember that the displacement is measured from the end of the two-byte instruction.)

Using JR in the inane program cited above, Figure 11-2 would be rewritten to look like Figure 11-5.

| | | |
|-------|-----|--------|
| 16514 | 62 | LD A,2 |
| 16515 | 2 | |
| 16516 | 60 | INC A |
| 16517 | 24 | JR,1 |
| 16518 | 1 | |
| 16519 | 60 | INC A |
| 16520 | 6 | LD B,0 |
| 16521 | 0 | |
| 16522 | 79 | LD C,A |
| 16523 | 201 | RET |

Figure 11-5. Substituting JR in Fig. 11-2

Unfortunately, there is no instruction for CALL RELATIVE. (Except for the telephone, of course.) I have encountered situations where I would have liked to have such an instruction available.

The complete list of JP and JR instructions will be tabulated in the next chapter.

That's all for now. There will be more.

by Tim Russell

This internal port will be wired in parallel with the right hand Timex joystick port. Although the same connector will be used, the two ports have independent operation. This means that any joystick plugged into J6 (the right socket) will be able to be read by means of the STICK command in 2068 BASIC, by the command IN 31 in 2068 or Spectrum BASIC, or by machine code I/O commands accessing port 1Fh (the Kempston port).

The circuit requires only three IC's, one pullup resistor pack, and two computer-type diodes. Looking



at the schematic, we can see that the 74LS02 and 74LS30 form the I/O address decoder, and the 74LS240 buffers and inverts the joystick data. The inversion is necessary because the Kempston standard calls for high-active output, and I needed low-active data at the joystick itself in order to make the parallel hookup. The last gate of the 74LS02 is used as an inverter to provide the base of Q7 with a "turn on" signal whenever the new port is accessed. To allow both signals (the Timex joystick port and the Kempston port) to access Q7 without damaging it, resistor R44 is disconnected at the A9 address connection, and one of the two diodes is added in series with it. Then, the second diode is connected from the R44/diode junction to the output of the 74LS02 "inverter." In order to provide the buffer with reliable data, all of the buffer inputs have pullup resistors.

For actual construction, I suggest a small piece of perfboard (about 2" by 1") as a mounting board. The 2068 has a conveniently empty area right by the right joystick port, which can be used for the board. PLEASE NOTE! I have absolutely no way of looking over your shoulders, and in NO WAY can I be responsible for any modifications which you choose to make to your computer! Therefore, BE CAREFUL!

I am not going to give exact details of the construction, partly because it is relatively simple, and partly because there may be a couple of

different 2068 printed circuit boards. Refer to the schematic whenever necessary, and use an ohmmeter to check for the proper signals.

To test your new toy, carefully reassemble your computer and turn it on. Test for proper operation of the Timex and Kempston joysticks with a looping line like that below:

```
10 PRINT AT 10, 10; STICK (1,2); AT 12, 10; STICK
(2,2): AT 14, 10; (IN 31): PAUSE 5: PRINT AT 10,
10; " "; AT 12, 10; " "; AT 14, 10; " ": GOTO 10
```

If the numbers you see on the screen match those below, you have a working Kempston/Timex joystick.

| | UPPER | MIDDLE | LOWER |
|-------|-------|--------|-------|
| UP | 1 | 0 | 8 |
| RIGHT | 8 | 0 | 1 |
| DOWN | 4 | 0 | 4 |
| LEFT | 4 | 0 | 2 |
| FIRE | 0 | 1 | 16 |

Remember that the numbers are additive as far as diagonal directions.

Well, that's the whole set of directions. I hope that many of you are able to use this circuit, and I invite any comments.

STRUCTURED BASIC TREE PROGRAMS

John C. Danielson

WHAT?

Structured BASIC is BASIC which is organized so that each module or section of a program has ONE entry point and ONE exit point. The opposite of this, so-called "spaghetti" BASIC, is programming that has multiple entry and/or exit points in its modules.

Structured BASIC also requires that no low-level module call (GOSUB to) a higher-level module. RETURNS to the module which called the routine are needed, and are naturally allowed. If you do not understand these statements, please continue-- as we explore structure, this point will become clear.

WHY?

Structured BASIC lends itself to easy maintenance and revision. This is true because the organized structure allows for rewriting one module without massive revisions elsewhere.

Further, it is easier to find and trace errors in branching if the program is structured. Not only that, but if each module is written as a single purpose module the location of logic errors becomes easier. Thus, structured BASIC is better because it is easier to debug, edit, and revise than "spaghetti".

Structure Explained

Take a look at Figure 1 and then return to read on.... If you turn Figure 1 upside down it will look like a simple tree.

This structure has been developed from database theory. Some databases have a "tree" design, and the lines indicate access routes or paths. For the purposes of programming, the lines indicate control routes designed into the program.

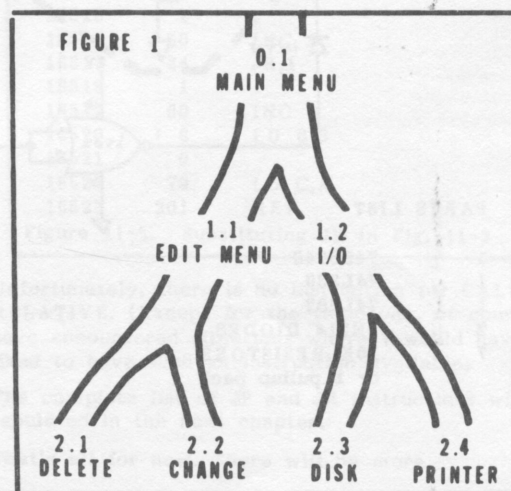


Figure 1 is simplified, in that each high level module only passes control to one of two lower level modules. In real life, each high level module could pass control to any one of several lower level modules. Further, each lower level module could pass control to any of several still lower level modules....

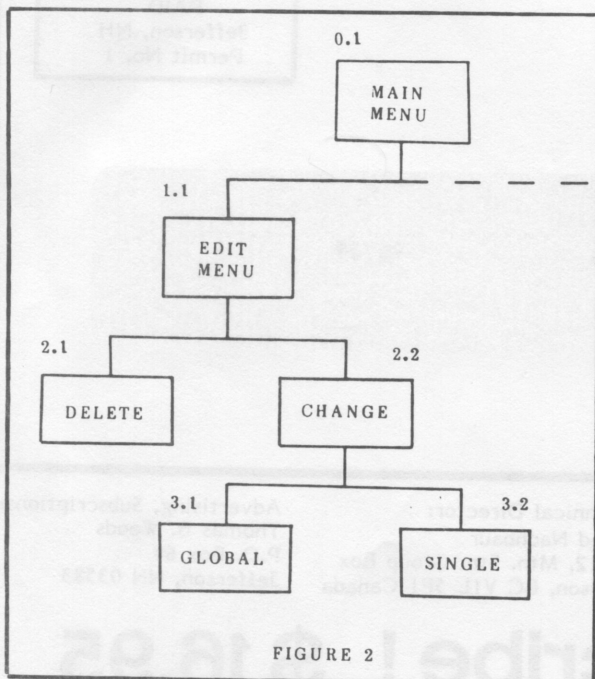
If you are confused at this point, look at Figure 1 and note that routine 0.1 could pass control to routine 1.1 or 1.2. If I had more space, I could have drawn in routines 1.3, 1.4, and 1.5 to the right of routine 1.2. Then, routine 0.1 could pass control to any one routine of the five (1.1-1.5). Get the idea?

Structure Charts Explained

A tool exists to show the structure of a program. The accepted name for this tool is "structure chart". Figure 2 presents a partial structure chart of a hypothetical word processing program. Take a look at it, and then continue.

Notice that the main menu (0.1) can pass control to the edit routine (1.1) and that the edit routine can then pass control to either delete (2.1) or change (2.2) routines. If the delete routine is chosen, it "does its thing" and then passes control back to the edit routine (1.1).

Now, pay close attention. If the delete routine were to pass control to routine 2.2 instead of passing back to the edit routine, you have half of a loop formed. If 2.2 passed back to 2.1 and 2.1 returned to 2.2 you would have a loop that could go on until you pressed the BREAK key. This would be POOR STRUCTURE, and very frustrating besides. That is why you see a nice series of vertical lines which meet the horizontal lines between each level. Please note here that the horizontal lines serve merely to show that the modules they each connect are on the same level, not that they are designed to call each other.



Structured Numbering

Structure charts have NO relation to line numbers assigned for each routine. However, some relationship between function and number would be handy if you plan to do a great deal of programming.

As an example, I use 10-100 for any DIM or LET statements needed for the program. Usually, 101-200 are reserved for the central program loop which houses the main menu and input routine(s). I also tend to use 7000-7099 for read utility routines, 8000-8299 for write utility routines, and 9000+ for termination routines. The numbers in between are used in order of level, from high to low. I also number in increments of 10 on the first program pass to allow lots of room for insertions due to minor logic gaps and errors. You may do as you please, but some system would make locating problems much simpler.

This was short, but I hope also sweet. We will explore more examples next time, when we will study the techniques of using a tool called a flow chart. For those of you not familiar with the term, flow charting outlines the details of logic you plan to use before you enter any BASIC at all. Why? Flow charts help to figure out what went wrong when the machine does not do what you expect.

John is 32 years old and resides in Michigan. He is currently seeking employment in the Data Processing field. His system is composed of a TS1000, Memotech CIF, dual Aerco disk drives, a letter quality and a dot-matrix printer, and 64K of RAM. This article was produced with disk-based Memotext.

SALE! 43% OFF!
WOW!

reg.
\$69.95

for ALL T/S
computers

**IN/OUT
PORTS**
SALE! \$39.95

Experiment with computer
sensing and control

The ideal port for sending signals from the outside world into your computer. Or use it to turn external devices on or off under computer control. 8 input lines, 8 outputs, plus 2 handshake strobes. Solderless female connectors. Now available at tremendous savings!

Thomas B. Woods
P.O. Box 64, Jefferson, NH 03583

CLASSIFIED

Do you have something to sell?
Let everyone know! Just \$2.75 a
line will put it here. 32 column
(screen) format, include spaces
in your text, and send CK or MO
with your listing.

Send to SWN Classified, 602 S.
Mill St., Louisville, OH 44641

Do color AND scaled tech
drawings with 2086 and ACCUDRAW!
Distance and angle readouts, two
display files, two undos, four
screen dumps, rubberbanding, 2-
speed cursor, fill, mirror, &
more. Use desk mount stick with
thumb-button or contact mouse.
Best with Aerco CIF. \$19.95 (+
1.05 CA tax)(includes S&H) M.O.,
check, or SASE to GLENN Technics
Box 2760, Anaheim, CA 92084

Don't forget the SWN Programmer's
contest. We have big prizes. Every
one has lots of fun. Don't be bashful.
Send us your entry. You have a good
chance to win super prizes, fame and
glory just from writing a silly little
program! Read all about how to enter
on page 10.

☆ SCLD ☆

Revive Your DEAD TS2068

OR

Make Your TS2068
INDESTRUCTIBLE!!

100 Chips- First come basis

\$20 ea. \$18 ea. for 2 or more

**W/instructions
optional 68PCC socket available**

cks to: CATS USER GROUP

Tom Bent
9016 Flicker Place
Columbia, MD 21045
Phone: (301) 730-7187

Chips Are HERE!!

SyncWare News

CONTENTS

| | |
|----------------------------------|----|
| For Your Support... | 2 |
| Forum... | 3 |
| Extensions to Pro/File Review... | 7 |
| Bennett | |
| Programmer's Contest!!! | 10 |
| Run Your Memotech 64K on | |
| the TS2068, Cook... | 11 |
| ABC/123 (TS1000/1500)... | 13 |
| Nachbaur | |
| Hang-Mult 2068... | 18 |
| Cook | |
| Basil's Compendium... | 19 |
| Wentworth | |
| One Chip Mod: Kempsonize | |
| Your Joystick Port... | 21 |
| Russell | |
| Structured Basic Tree | |
| Programs, Danielson... | 22 |

FIRST-CLASS MAIL
U.S. POSTAGE
PAID
Jefferson, NH
Permit No. 1

Z.A. Lewis
PO Box 559
Camino, CA

95709

4:4

Editor:
Jeffrey Moore
602 S. Mill Street
Louisville, OH 44641

Technical Director:
Fred Nachbaur
C-12, Mtn. Stn. Group Box
Nelson, BC V1L 5P1 Canada

Advertising, Subscriptions:
Thomas B. Woods
P.O. Box 64
Jefferson, NH 03583

Submissions, announcements, letters to the editor,
and all other material of editorial interest should
be sent to the Ohio address.

Copyright 1986 SyncWare News

Subscribe ! \$ 16.95

For Canada and Mexico please add \$3.00 postage.
All other foreign add \$9.00 Airmail (U.S. funds only)

1 year (6 issues)